UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN

ESPECIALIDAD
SISTEMAS DE TELECOMUNICACIÓN

PROYECTO FIN DE CARRERA

HERRAMIENTA DE VISUALIZACIÓN Y ANÁLISIS DE LA LOCALIZACIÓN DE VECTORES SOPORTE EN RECONOCEDORES DE HABLA HÍBRIDOS HMM/SVM

Autor: EMILIO DE OTEO PÉREZ
Cotutores: RUBÉN SOLERA UREÑA
FERNANDO DÍAZ DE MARÍA

FEBRERO DE 2009

Proyecto Fin de Carrera
HERRAMIENTA DE VISUALIZACIÓN Y ANÁLISIS DE LA LOCALIZACIÓN DE VECTORES SOPORTE EN RECONOCEDORES DE HABLA HÍBRIDOS HMM/SVM

Autor
EMILIO DE OTEO PÉREZ

Cotutores
RUBÉN SOLERA UREÑA
FERNANDO DÍAZ DE MARÍA

La defensa del presente Proyecto Fin de Carrera se realizó el día 17 de Febrero de 2009, siendo calificada por el siguiente tribunal:

PRESIDENTE: ASCENSIÓN GALLARDO ANTOLÍN

SECRETARIO: CARMEN PELÁEZ MORENO

VOCAL: ISAAC SEOANE PUJOL

y habiendo obtenido la siguiente calificación:

CALIFICACIÓN: NOTABLE

Leganés, a 17 de Febrero de 2009

Agradecimientos

Gracias Rubén y Fernando, sin vuestra ayuda y paciencia la tinta que mancha estas páginas no hubiera salido nunca de la impresora.

A mis padres, Ramón y Angelines, por haberme dado la oportunidad de conseguir esto y tantas otras cosas.

A mis hermanos, Santi, Francisco y Sagrario y mis cuñadas Ana y Ángela, por haber sido la mano en la espalda que te impide dar la vuelta y te anima a seguir caminando.

A mis sobrinos, Laura, Adrián y Natalia, por enseñarme que hay baterías que no se agotan nunca.

A los torrijeños, Álvaro, Samu, Juan Pedro, Rodrigo, Fran, Manu, Ramírez, More, Ronce, Palomo, Juan Carlos, Rober, Nacho, Diego, Marta, Elisa, Mª Ángeles, Arantxa, Estrella, y Cristina, por ser mucho más que unos amigos.

A los de "la Charly", Jose, Nacho, Pablo, Javi, Flori, Isa, Noelia R., Noelia L., Sara, Fátima, Mariu,…, por regalarme vuestra amistad y todos los buenos momentos que hemos pasado juntos y los que nos quedan por pasar.

A los del piso, Abilio, Alberto, Héctor, Emilio y Juan, por los risk, los cines, las cervezas, las risas, los macarrones, las anécdotas…

Y a "los Telefónicos", en especial a "los Tariferos" (perdonadme pero no cabéis todos), por darme la oportunidad de trabajar a vuestro lado.

A todos vosotros, por ser "cachitos" de mi vida, GRACIAS.

“Fuerza y honor”

Resumen

Este Proyecto Fin de Carrera está enmarcado en el campo del reconocimiento automático de habla (RAH) mediante reconocedores híbridos, los cuales aprovechan las propiedades de las distintas tecnologías que lo constituyen. En este ámbito se aborda la aplicación conjunta de los modelos ocultos de Markov (HMMs) y las máquinas de vectores soporte (SVMs) para el reconocimiento automático de habla robusto. Se entra, por tanto, en el marco de los reconocedores de habla híbridos, donde la hibridación consistirá en sustituir la etapa de modelado acústico tradicional de los HMMs, modelos de mezclas de Gaussianas (GMMs), por una SVM. En ésta última, la decisión depende de los denominados vectores soporte (SVs), muestras de entrenamiento que intervienen en la decisión y sobre las cuales se centrará la investigación de este Proyecto.

El objetivo de este Proyecto Fin de Carrera es realizar un estudio acerca de la localización de los SVs, con el fin de determinar si estos se localizan en algún punto concreto de los fonemas, ya sea en sus transiciones o en sus partes estables. Para ello se ha diseñado una interfaz gráfica en Matlab que nos ayudará al estudio y la localización visual de estos vectores soporte dentro de los distintos ficheros de audio que componen la base de datos. La herramienta desarrollada permite varios tipos de representaciones gráficas, combinando la señal de voz en los dominios del tiempo y la frecuencia con los SVs de una clase determinada o todos los que pertenezcan a esa señal. Además, muestra como referencias las transiciones entre los distintos fonemas para facilitar una mejor interpretación de los elementos representados.

El análisis ha sido completado con un estudio estadístico acerca de la relación existente entre la localización de los distintos SVs y las transiciones entre fonemas. Dichas transiciones se han obtenido mediante varios métodos de segmentado de la señal y se usarán como referencia para aportar una conclusión cuantitativa al estudio.

Índice

Lista de Figuras

Lista de Tablas

Glosario

RAH: Reconocimiento/Reconocedor Automático de Habla (en inglés: ASR, Automatic Speech Recognition).

HMM: Hidden Markov Model (en español: Modelo Oculto de Markov).

SVM: Support Vector Machine (en español: Máquina de Vectores Soporte).

GMM: Gaussian Mixture Model.

SV: Support Vector (en español VS: Vector Soporte).

HTK: Hidden Markov Model Toolkit.

BBDD: Base de Datos.

PC: Personal Computer.

PBX: Private Branch Exchange, central telefónica conectada a la red pública.

PDA: Personal Digital Assistant.

DTW: Dynamic Time Warping.

LPC: Linear Predictive Coding.

DTFT: Discrete Time Fourier Transform.

MFCC: Mel-Frequency Cepstrum Coefficient.

DCT: Discrete Cosine Transform.

ANN: Artificial Neural Network.

TDNN: Time-Delay Neural Network.

RNN: Recurrent Neural Network.

MLP: Multilayer Perceptron.

EM: Expectation-Maximization.

RBF: Radial Basis Function.

GUI: Grafic User Interface.

Capítulo 1.
Introducción

Un reconocedor automático de habla es un sistema capaz de transcribir en texto una locución sonora. Por lo tanto este sistema debe ser capaz de reconocer la voz humana, adaptándose a la gran variabilidad presente en ésta (entonación, velocidad, vocalización, ruido…).

En la actualidad, las aplicaciones que incorporan RAH son usadas en multitud de ámbitos (robótica, atención telefónica, ocio y entretenimiento); esto unido a que los dispositivos electrónicos capaces de albergar sistemas de reconocimiento de habla evolucionan hacia la miniaturización (teléfonos móviles, ordenadores portátiles, videoconsolas…), lo que en la mayoría de los casos va ligada a una disminución de los recursos disponibles (capacidad de procesador, memoria…), desemboca en una reducción en las tasas de reconocimiento. Por lo tanto, buscaremos que las prestaciones de los RAH sean las mejores independientemente del contexto ambiental en el que estemos, es decir, pretenderemos obtener reconocedores automáticos de habla robustos.

1.1. Problema.

En esta situación, nos encontramos ante la necesidad de aumentar la robustez de los reconocedores de habla, para adecuarlos a las nuevas aplicaciones, entornos y plataformas en los que se deben integrar. Este Proyecto Fin de Carrera se centra, dentro de las distintas tecnologías de RAH empleadas en la actualidad, en los reconocedores híbridos. Estos sistemas combinan las ventajas de los sistemas de reconocimiento automático de habla tradicionales basados en HMMs con otras técnicas, que aportarán una mayor fiabilidad en entornos ruidosos, ya que éste es uno de los puntos donde los sistemas tradicionales basados en HMMs se muestran menos robustos.

La aplicación de los modelos ocultos de Markov a las tareas de RAH supuso una revolución en este ámbito, pero los HMMs muestran ciertas limitaciones cuando deben trabajar en escenarios reales, donde un factor muy influyente es el ruido. Dado que el margen de mejora de los sistemas basados en HMMs es reducido, debido a que es una tecnología sobre la que se ha trabajado de forma exhaustiva, una opción viable es el uso de sistemas híbridos. Estos sistemas muestran resultados prometedores pero requieren aún un mayor desarrollo. El híbrido que se usará en este Proyecto Fin de Carrera mantiene la estructura tradicional de los RAH basados en HMMs y deja la responsabilidad del modelado acústico en manos de una máquina de vectores soporte, en principio más robusta ante ruido e interferencias que los tradicionales modelos de mezclas de Gaussianas (GMMs), encargados de realizar esta tarea en los sistemas tradicionales basados en HMMs.

1.2. Objetivos.

En este Proyecto Fin de Carrera se parte de un sistema híbrido HMM/SVM en el que se utiliza una SVM en la etapa de modelado acústico. Será dicha SVM la que nos proporcione la probabilidad a posteriori de cada unidad acústica considerada para una observación dada. Dichas probabilidades se calculan a partir de la salida blanda de la SVM, la cual depende de los vectores soporte. Por lo tanto, es especialmente interesante analizar la posición de dichas muestras dentro de los fonemas presentes en la base de datos de entrenamiento. Por lo tanto, el objetivo de este PFC es extraer los SVs resultantes del entrenamiento de la máquina, determinar su posición dentro del fichero correspondiente de la base de datos y representarlos de forma que nos aporte la mayor información posible. Para ello no solo es necesario conocer su posición dentro de un determinado fichero, sino que necesitamos su ubicación dentro del fonema correspondiente. Por consiguiente, es necesario disponer de las transiciones entre los fonemas de cada fichero que compone la base de datos, ya que nos proporcionarán una referencia sobre la posición de los SVs. Como se verá en el capítulo 5 de este PFC, se emplean varios métodos para obtener dichas transiciones.

Para llevar a cabo el estudio estadístico, uno de los hitos del proyecto es el desarrollo de una herramienta que nos permita, entre otras cosas, trabajar de forma visual con los vectores soporte, en los dominios del tiempo y la frecuencia, donde además contaremos con la referencia de las transiciones entre fonemas mencionadas en el párrafo anterior. Los datos derivados de este análisis se estudiarán con la intención de aportar ciertas conclusiones acerca de la posibilidad de establecer alguna relación entre los SVs y su localización dentro de los distintos fonemas.

.1.3. Planteamiento.

El trabajo se ha estructurado de la siguiente manera:

- Se parte de una base de datos compuesta por 160.000 locuciones, grabadas en ambientes sin ruido o con ruido controlado (oficina) y por 4.000 locutores distintos. De esta base de datos se selecciona un subconjunto para facilitar la realización de los experimentos, dado el elevado coste computacional de la SVM.
- Dada la anterior base de datos, se entrenan tanto los HMMs como las SVMs que componen el reconocedor híbrido empleado. Para realizar el entrenamiento y reconocimiento de los HMMs nos ayudaremos de las herramientas que proporciona el paquete de aplicaciones HTK [1]. Para entrenar la SVM se usa el software LibSVM [2]. De este entrenamiento se obtienen los SVs asociados a cada una de las unidades acústicas consideradas, que en nuestro caso serán los 33 fonemas que componen el diccionario de la BBDD.
- Se extraen los SVs que componen el modelo de la SVM y se hallan las muestras de la base de datos a las que corresponden y su localización dentro de ellas.
- Se lleva a cabo el proceso de reconocimiento propiamente dicho; de este proceso se obtendrán las transiciones determinadas por el RAH híbrido.
- Se realiza un segmentado manual de la base de datos para obtener una referencia en la ubicación de los SVs respecto a las transiciones entre fonemas. Para la realización de este segmentado manual se ha empleado la herramienta de edición de audio Cool Edit.

- Basándonos en Matlab, se desarrolla una interfaz gráfica que nos permita una visualización de la localización de los vectores soporte sobre los correspondientes ficheros de voz.
- Por último, una vez recopilada toda la información de los pasos anteriores se realizará el estudio estadístico respecto al segmentado manual y el segmentado del reconocedor automático de habla híbrido HMM/SVM, intentando hallar algún patrón en la localización de los SVs.

1.4. Estructura de la memoria.

A continuación se presenta la estructura que da forma a la memoria de este Proyecto Fin de Carrera:

Capítulo 1. Es una introducción al contenido de la memoria donde se resumen los problemas que motivan este proyecto, los objetivos que se buscan y el planteamiento que se sigue para lograrlo.

Capítulo 2. Este capítulo nos introduce en el ámbito de los reconocedores automáticos de habla, presentando sus principales usos y algunas aplicaciones comerciales reales. Se tratan también dentro de este capítulo los obstáculos a los que se enfrenta, así como las tecnologías más comunes en el reconocimiento automático de habla. Concluye el capítulo particularizando sobre los RAH basados en modelos ocultos de Markov, tratando de forma individual cada una de las partes que componen un RAH y qué tecnologías se aplican en cada una de ellas.

Capítulo 3. En él se desarrollan los sistemas híbridos, tratando en primer lugar los híbridos con redes neuronales y posteriormente la hibridación con máquinas de vectores soporte. Se profundizará en los fundamentos matemáticos de estos últimos (híbridos HMM/SVM) ya que será la tecnología que se emplea en este proyecto fin de carrera. Para cerrar este capítulo se hará una introducción a la herramienta gráfica desarrollada.

Capítulo 4. Continuaremos en este capítulo con la descripción de la base de datos empleada en este PFC. Se analizará el entrenamiento del híbrido HMM/SVM realizado con esta base de datos y cómo se obtuvieron los vectores soporte elegidos para componer el modelo del híbrido. El capítulo finaliza con la descripción de las características de la herramienta gráfica implementada para el análisis de estos SVs, tratando tanto los elementos que la componen, como los tipos de representaciones que ofrece y detallando las características de cada una de ellas.

Capítulo 5. En este capítulo se presentan los resultados obtenidos en base a los distintos tipos de segmentado realizados sobre la base de datos para determinar las transiciones entre fonemas. Con estos resultados se tratará de inferir alguna tendencia en la ubicación de los SVs.

Capítulo 6. Llegados a este punto se recopilarán los datos obtenidos y serán analizados junto con los datos expuestos en el resto de capítulos de la memoria, para aportar las conclusiones finales sobre el estudio realizado en este PFC.

Capítulo 7. Como cierre de esta memoria se presentará en un presupuesto, la valoración económica así como la descomposición y descripción de las tareas seguidas en la elaboración de este PFC.

Como *Anexos* se incluyen datos e información considerados de interés y que aportan valor al conjunto de la memoria.

Capítulo 2.
Reconocimiento Automático de Habla

La finalidad que persiguen los reconocedores de habla es conseguir una comunicación más natural y fluida entre los seres humanos y las máquinas [3], permitiendo una mayor interactividad entre ambos. Para ello, un sistema de RAH debería estar dotado idealmente de un sistema lo más parecido al oído humano, funcionalmente hablando, que sea capaz de detectar y capturar el habla de una persona y procesarla de tal manera que pueda llegar a diferenciar cada una de las palabras pronunciadas por el locutor e interpretarlas dentro del contexto en el que se encuentre.

A continuación se tratarán la estructura de los RAH y las tecnologías que emplean actualmente con el objetivo de conseguir esta interacción hombre-máquina.

2.1. Estado del Arte.

Necesidad del Reconocimiento Automático de Habla:

Día a día las máquinas se están haciendo mucho más presentes en la vida de los seres humanos (en el hogar, en el trabajo, en el ocio…). Es lógico pensar, por tanto, que el hombre trate de comunicarse con ellas al igual que lo hace con otro ser humano, es decir, mediante el habla, que es el modo natural de comunicación entre seres humanos. Esto hace que los sistemas de RAH deban estar presentes en todas las máquinas con las que se desee interactuar de esta manera. Las condiciones en las que se producirá la comunicación entre ambos podrán estar sujetas a una gran variabilidad, por lo que las tecnologías empleadas en los RAH variarán en función del propósito para el que se destinen.

A continuación se describen algunos de los escenarios típicos de aplicación de los sistemas de RAH:

- Dictado automático: En los últimos años se ha convertido en el uso más común de las tecnologías de reconocimiento de voz. Ha adquirido mucho peso en la redacción de informes, tanto médicos, como legales, además de emplearse en la redacción de artículos periodísticos.

- Búsqueda y recuperación de información en bases de datos (textuales y sonoras): Con estos mecanismos se podrían recuperar fragmentos de una base de datos, realizando búsquedas por determinas palabras o frases clave.

- Identificación y verificación de locutor: Aunque estos sistemas no usan los RAH como tecnología fundamental, sí se apoyan en ellos para realizar estas tareas, por ejemplo, mediante el uso de una 'firma vocal', con la que realizar gestiones legales o transacciones bancarias mediante el teléfono.

- Control por comandos: Sistemas de reconocimiento de habla diseñados para dar órdenes a un computador. Actualmente se usan para interactuar con el sistema operativo de un PC, manejar robots, operar los accesorios en un coche, el control de los sistemas domóticos en el hogar… Estos sistemas reconocen un vocabulario muy reducido, lo que permite altas prestaciones.

- Telefonía: Algunos sistemas PBX permiten a los usuarios ejecutar comandos mediante el habla, en lugar de pulsar tonos. En muchos casos se pide al usuario que diga un número para navegar en un menú o datos personales con vocabularios más o menos reducidos y controlados.

- Dispositivos portátiles: Los dispositivos portátiles de pequeño tamaño, como las PDAs o los teléfonos móviles, tienen unas restricciones muy concretas de tamaño y capacidad, lo que dificulta el uso de otros tipos de interfaces de acceso más voluminosos, por lo que el habla podría ser una alternativa adecuada.

- Sistemas diseñados para discapacitados y ayuda logopédica: Estos sistemas no sólo posibilitan una mayor accesibilidad para las personas discapacitadas (hacen que personas con dificultad para teclear puedan manejar un PC con la voz o que personas con dificultades auditivas puedan leer la trascripción de una conversación telefónica), sino que también pueden llegar a ser una buena herramienta para ayudar a personas con problemas de habla.

Una vez enunciados algunos de los ámbitos típicos de aplicación, pasamos a detallar algunos de los problemas a los que se enfrenta el RAH, cuando se expone a escenarios reales.

Problemas a los que se enfrenta:

El reconocimiento de habla tiene que ser efectivo para cualquier tipo de locutor y en el mayor número de ambientes y situaciones posibles, por lo que los sistemas de RAH deben enfrentarse a los siguientes problemas principalmente [4]:

- Variación inter-locutor: son las variaciones a las que se deben enfrentar los sistemas de RAH y que se deben a la heterogeneidad en el conjunto de locutores que componen la base de datos. Los rasgos más relevantes que constituyen las variaciones inter-locutor son el sexo, la edad, la procedencia geográfica y el físico del locutor, entre otros.

- Variación intra-locutor: los sistemas de RAH no sólo se deben enfrentar a una gran variabilidad de locutores, sino que además para un mismo locutor el proceso de reconocimiento vocal se puede ver afectado por diversos factores que afecten al habla del individuo, como pueden ser su estado anímico o estado físico, condiciones que el sistema de RAH debe ser capaz de solventar en la medida de lo posible.

- Variación en la velocidad de locución: no todas las personas tienen la misma velocidad hablando, ni hacen las pausas de la misma duración. Por lo tanto este es otro de los factores que debe tratar de compensar un buen reconocedor de habla.

- Reconocimiento palabras aisladas/habla continua: a la hora de diseñar un sistema de RAH, uno de los factores que tiene un gran peso es si este sistema deberá reconocer palabras aisladas (ya sean de un vocabulario reducido o más amplio) o si por el contrario debe ser capaz de reconocer el habla continua.

- Amplitud del vocabulario: cuanto más reducido sea el vocabulario (elementos que debe reconocer el sistema) con el que debe trabajar el RAH, menos complejidad tendrá éste.

- Sistemas dependientes/independientes del locutor: los sistemas dependientes son aquellos que son específicamente diseñados para ser utilizados por un locutor concreto, por lo que sus prestaciones suelen ser mayores que las de los sistemas independientes, diseñados para ser usados por múltiples locutores, donde entrarían en juego las variaciones intra/inter locutor mencionadas anteriormente.

- Ruido: como en todo sistema, el ruido es un factor determinante que incrementa de forma considerable la tasa de error, disminuyendo en nuestro caso la tasa de reconocimiento. Como es lógico, no es lo mismo reconocer habla grabada en un estudio aislado y en condiciones óptimas de calidad, que hacerlo sobre locuciones grabadas en cualquier otro entorno, donde pueden aparecer múltiples tipos de ruido o interferencias. Este es el principal problema que se aborda con los sistemas híbridos basados en SVMs, con las que se busca una mayor robustez.

Ejemplos de aplicaciones comerciales reales:

En la actualidad existen en el mercado diversos productos de reconocimiento de voz que se adaptan a algunas de las necesidades que se describieron anteriormente. Algunos ejemplos son:

- Speech Magic (Philips Speech Recognition Systems): sistema de dictado desarrollado por Philips y actualmente enfocado a la redacción de cualquier tipo de informe, especialmente médicos [5].

- Via Voice de IBM: sistema de reconocimiento de habla y conversión texto habla para su uso en dispositivos portátiles, manos libres y control de accesorios en vehículos [6].

- Dragon Naturally Speaking de Nuance: sistema de dictado que entre otras aplicaciones permite controlar con la voz las aplicaciones de Windows, realizar búsquedas en ficheros de audio …[7]

- Telefónica: Software vocal de Telefónica usado, entre otros, en sistemas de atención telefónica a usuarios.

- Telisma: ofrece servicios de atención telefónica automática mediante su producto TeliSpeech [8].

También se pueden encontrar algunos programas gratuitos como:

- PerlBox: es un gestor de sistema operativo por comandos de voz [9].
- Sphinx, del Sphinx Group en Carnegie Mellon University [10].
- HTK: es el software que se ha empleado en este PFC y se compone de un paquete de librerías para el desarrollo de sistemas de RAH basados en HMMs. Proporciona una serie de herramientas que permiten realizar tanto el entrenamiento de los HMMs como el reconocimiento [1].

Tecnologías empleadas:

Históricamente las primeras tecnologías que se emplearon en el RAH fueron las basadas en el r*econocimiento de patrones.* Estas técnicas se basan en la comparación de la locución que se desea reconocer con una serie de patrones de referencia que representan a cada una de las unidades que componen el diccionario del sistema. La locución de entrada será asignada al modelo cuyo patrón sea más "parecido" a ella. Un ejemplo es el Dynamic Time Warping [11], que incluye un alineamiento temporal necesario frente a variaciones en la velocidad de locución. Estas técnicas fueron las pioneras gracias a su simplicidad, pero llevan aparejado un gran coste computacional, ya que cada muestra de entrada deberá compararse con todos los patrones que

caracterizan las distintas unidades del diccionario. Este gran coste computacional hizo que se buscaran técnicas alternativas, una de las cuales fueron los *modelos ocultos de Markov [12]*.

Los HMMs modelan cada unidad acústica que se desea reconocer mediante un autómata de estados finitos, donde en cada instante de tiempo el modelo puede cambiar a otro estado o permanecer en el mismo. Este cambio está determinado por una probabilidad de transición dada. Los HMMs son una extensión de los modelos de Markov, donde la observación que se produce en cada estado está ligada a un proceso probabilístico, no observable de forma directa, de ahí su adjetivo "ocultos". La principal ventaja que presentan es que son capaces de trabajar de una forma muy elegante con locuciones de distinta longitud, debido a su estructura de estados. Estos modelos requieren de una fase previa de entrenamiento, donde los modelos de mezclas de Gaussianas se entrenan de forma iterativa, consiguiendo que el modelo resultante sea lo más general posible.

En los últimos años y gracias al aumento en la capacidad de cómputo de los sistemas actuales, ha ganado importancia una nueva técnica de reconocimiento de habla basada en las *redes neuronales.* Éstas se han ido incorporando como técnicas de reconocimiento automático de habla, consiguiendo en algunos casos resultados similares a los HMMs. Su principal virtud es su capacidad de aprendizaje discriminativo pero, por el contrario, su entrenamiento es algo más costoso en términos de tiempo que el de los HMMs. Otro factor que pesa en su contra es el desconocimiento sobre el número necesario de nodos que se debe emplear y de la estructura de capas. Además puede llegar a dar resultados no óptimos si el entrenamiento se detiene en un mínimo local de la función de coste. Su uso principal aparece en sistemas híbridos HMM/ANN, aunque también se pueden usar otras tecnologías en estos sistemas híbridos, usando por ejemplo *máquinas de vectores soporte*.

Este proyecto fin de carrera se centrará en los RAH basados en modelos ocultos de Markov y en los sistemas híbridos de reconocimiento de habla HMM/SVM, que proporcionan una mayor robustez frente al ruido. Para ello se partirá de un sistema basado en HMMs donde los modelos de mezclas de Gaussianas que componen el modelado acústico serán reemplazados por las SVMs.

2.2. Reconocimiento Automático de Habla basado en HMMs.

Los HMMs son la tecnología de reconocimiento automático de habla más empleada. Un HMM es una máquina de estados, donde en su modelo general (Figura 1) todos sus estados se encuentran interconectados entre si por unas determinadas probabilidades de transición a_{ij}, y cada estado genera un vector de observación de acuerdo a una determinada función de densidad de probabilidad. La salida del modelo, de tipo probabilística, es función de la secuencia de estados recorrida, también probabilística. Esto es lo que hace que se denominen "ocultos", ya que no podemos observar una determinada secuencia de estados, sino que sólo sabemos que se ha podido recorrer una secuencia concreta de estados con una probabilidad determinada.

Un HMM genérico estará definido por N estados, una matriz de transiciones entre estados A y las funciones de densidad de probabilidad de emisión asociadas a cada estado B. En la figura siguiente se representa un HMM de tres estados. Se nombrará como $q_t = N$ a la probabilidad de que en el instante t el modelo se encuentre en el estado N.

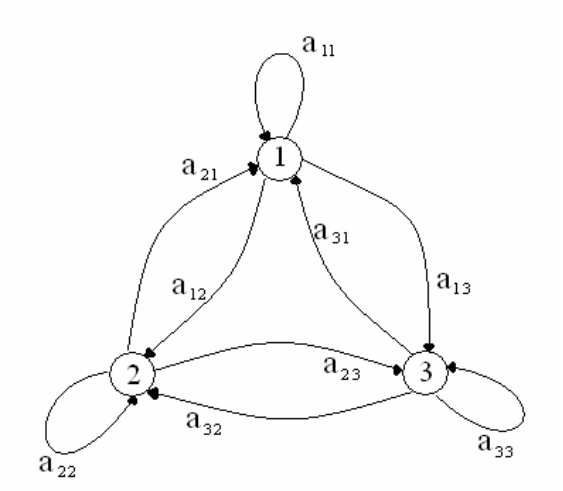


Figura 1. HMM Genérico de 3 Estados

Las transiciones entre estados están definidas por la matriz de transiciones (en adelante A). Para el caso particular de la Figura 1 se tiene una matriz de transiciones:

$$A = \begin{Bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Bmatrix} \tag{2.1}$$

Generalizando para N estados tendríamos:

$$A = \{a_{ij}\}, 1 \le i, j \le N \tag{2.2}$$

donde cada uno de los a_{ij}, denota la probabilidad de que estando en el estado i en un determinado instante de tiempo, se pase al estado j en el siguiente:

$$a_{ij} = P[q_{t+1} = j \mid q_t = i] \tag{2.3}$$

Por otro lado, la probabilidad de observar o_t en el estado j vendrá dada por una función de densidad de probabilidad que denotaremos de la siguiente forma:

$$b_j(o_t) = P[o_t \mid q_t = j] \tag{2.4}$$

donde o_t será la observación en el instante t.

El último de los elementos que definirá el modelo es el vector de inicialización, llamado π.

$$\pi = \{\pi_i\}, 1 \le i \le N \tag{2.5}$$

$$\pi_i = P[q_1 = i] \tag{2.6}$$

donde π_i denota la probabilidad de comenzar la secuencia de observación en cada uno de los i estados.

Estos son los parámetros que definen cualquier HMM y su agrupación se expresa como $\lambda = (A, B, \pi)$. Los HMMs son modelos generativos que se pueden emplear para generar una secuencia de observaciones $O = O_1 O_2 O_3 O_4 O_5 ... O_T$, o para calcular la probabilidad de haber generado esa secuencia de observaciones con ese modelo.

Los HMMs admiten multitud de configuraciones, cada una de las cuales tendrá diferentes propiedades. Un primer paso en el diseño de un reconocedor de habla basado en HMM será elegir la topología apropiada. En reconocimiento de habla, la topología de HMM más común es la denominada "izquierda-derecha", en la cual no se puede volver desde un estado a los anteriores. También suelen emplearse modelos de primer orden, en los que de un estado sólo se puede pasar al siguiente o permanecer en el mismo.

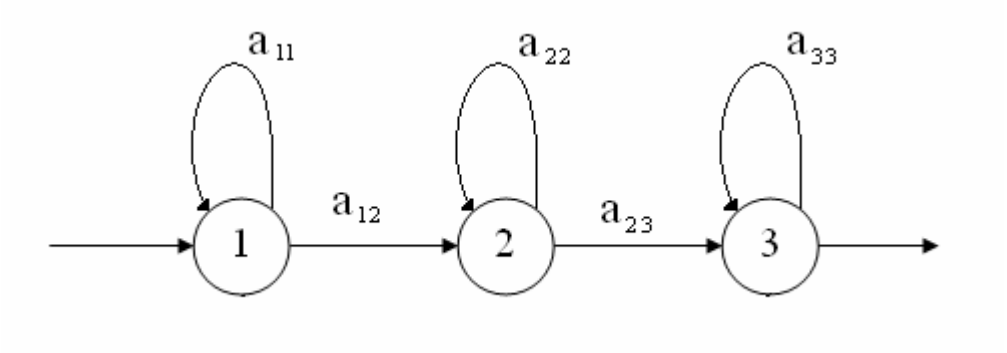


Figura 2. HMM "izquierda-derecha" de primer orden con 3 estados.

Al hablar se van enlazando fonemas, de forma que un fonema puede verse modificado en función de los fonemas que lo rodean (este efecto es conocido como coarticulación). Esto ha llevado a distinguir habitualmente tres partes diferenciadas en cada fonema. La primera y la tercera estarán condicionadas por los fonemas anterior y posterior, respectivamente. La segunda comprende la parte estable del fonema. Por lo tanto parece intuitivo asociar a cada fonema un HMM de tres estados. Esto no descarta el uso de modelos con más de tres estados en casos concretos.

De todo lo anterior se deriva que un reconocedor de habla basado en HMMs deberá estar compuesto por un modelo por cada unidad acústica que se desee reconocer. En el caso particular de este Proyecto, el diccionario está compuesto por 33 fonemas.

2.3. Estructura de un Reconocedor Automático del Habla.

A continuación se detallan los bloques fundamentales que componen cualquier sistema de RAH, cuyo diagrama se muestra en la Figura 3:

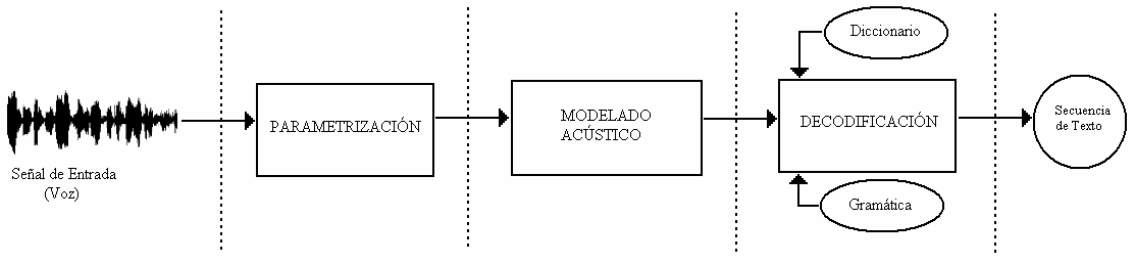


Figura 3. Esquema de un RAH.

2.3.1. Parametrización.

Esta fase debe ser capaz de extraer características representativas de la señal de voz y, más en concreto, de la unidad básica con la que se trabaje, de forma que posteriormente ésta pueda ser fácilmente identificada y reconocida. De esta fase depende gran parte del éxito de cualquier reconocedor, ya que si no somos capaces de encontrar parámetros que representen de una forma discriminativa las unidades con las que tratamos, difícilmente se podrán distinguir unas de otras en la etapa de decodificación [13].

En la actualidad son varias las parametrizaciones que se usan, pero las más comunes son la LPC y los coeficientes Mel-Cepstrum. Todas estas parametrizaciones tratan de obtener una representación de la envolvente espectral de la señal, que es la información más relevante para el reconocimiento.

- *LPC [14]*: La parametrización LPC asume un modelo antirregresivo para la voz. Si llamamos $s[n]$ a la señal de voz y $e[n]$ a la excitación, tenemos:

$$s[n] = \sum_{k=1}^{p} a_k \cdot s[n-k] + e[n] \qquad (2.7)$$

donde a_k son los coeficientes de predicción lineal.

De aquí podemos obtener la función de transferencia que modela el efecto del tracto vocal:

$$H(z)=\frac{S(z)}{E(z)}=\frac{1}{1-\sum_{k=1}^{p}a_k\cdot z^{-k}}=\frac{1}{A(z)} \qquad (2.8)$$

Los coeficientes a_k se estiman minimizando el error cuadrático medio sobre la trama de voz. Para ello se deberá resolver un sistema de p ecuaciones con p incógnitas por el método de la autocorrelación.

- *Coeficientes Cepstrum – Mel-Cepstrum*: El Cepstrum se define como la transformada de Fourier inversa del logaritmo del módulo del espectro de la señal de voz.

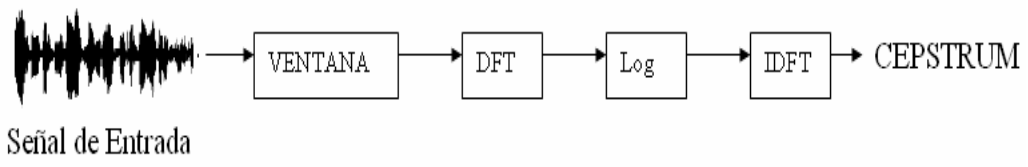


Figura 4. Diagrama generación Cepstrum.

La voz se puede descomponer como la combinación de dos contribuciones distintas. Una de ellas es una componente periódica debida a la vibración de las cuerdas vocales que produce los sonidos sonoros. La otra es una fuente de ruido aleatorio, que se encarga de producir los sonidos sordos. La combinación de estas dos señales es posteriormente modulada por la lengua, dientes y labios, dando como resultado los distintos fonemas.

El objetivo de esta parametrización es desacoplar las componentes de excitación ($G(\omega)$) y del tracto vocal ($Y(\omega)$) presentes en el espectro de la voz ($X(\omega)$), que se puede representar como:

$$|X(\omega)|=|G(\omega)|\cdot|Y(\omega)| \qquad (2.9)$$

Si se aplican logaritmos se obtiene:

$$\log|X(\omega)| = \log|G(\omega)| + \log|Y(\omega)| \qquad (2.10)$$

Posteriormente se aplica la transformada de Fourier inversa. Debido a la distinta tasa de variación espectral de las dos componentes, estas quedan separadas en el nuevo dominio "cepstral", por lo que ya es posible separar ambas componentes aplicando el filtro adecuado.

Los coeficientes cepstrum se calculan como:

$$c(n) = \frac{1}{2 \cdot \pi} \cdot \int_{-\pi}^{\pi} \log|X(\omega)| \cdot e^{j\omega n} \cdot d\omega \qquad (2.11)$$

Una ventaja de los coeficientes cepstrum es que están decorrelados, lo que nos permite emplear matrices de covarianzas diagonales en los HMM. Esto reduce el número de parámetros que hay que estimar en la fase de entrenamiento y, por tanto, su complejidad.

Algunos de los parámetros más importantes que se suelen derivar de los coeficientes cepstrum y que se usan combinados con ellos son:

- Cepstrum diferenciales: aportan información sobre las variaciones dinámicas del espectro y la trayectoria de los formantes a lo largo de la locución. Son las diferencias temporales de los coeficientes cepstrales.

- Energía diferencial: similar a los cepstrum diferenciales, pero aplicado a la energía.

En problemas de RAH es habitual el uso de los coeficientes Mel-Cepstrum (MFCC). Para ello la DTFT de la señal de voz pasa por un banco de filtros triangulares solapados y centrados sobre una escala no lineal (MEL):

$$m = 2595 \cdot \log_{10}\left(1 + \frac{1}{700}\right) \qquad (2.12)$$

Posteriormente, se aplica el logaritmo a la salida del banco de filtros y se calcula la DCT de los coeficientes:

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} \log(m_j) \cdot \cos\left[\frac{i \cdot \pi}{N} \cdot (j - 0.5)\right], \quad i = 0, \ldots, N_c \qquad (2.13)$$

donde:

N: número de filtros del banco

m_j: salida del j-ésimo filtro

N_c: número de coeficientes Mel-Cepstrum deseado.

De forma habitual, c_0 es reemplazado por la log-energía de la trama.

En lo que respecta a este proyecto, la parametrización empleada ha sido la MFCC, dado que en la actualidad es la más común. Se han obtenido 12 MFCCs, la log-energía, 13 diferencias primeras (cepstrum diferenciales + energía diferencial) y 13 diferencias segundas. Los 12 coeficientes MFCCs han sido normalizados en media fichero a fichero y la log-energía se normaliza de tal forma que su valor máximo en el fichero sea 1.

2.3.2. Modelado Acústico.

Si seguimos el esquema planteado para un RAH basado en HMMs, la fase del modelado acústico será la que nos proporcione las probabilidades de emisión ($b_j(\bar{o})$) de cada estado.

La fase del modelado acústico en los HMMs se basa en los GMMs, cuyos parámetros se deben haber obtenido previamente mediante el entrenamiento del sistema. En los sistemas tradicionales basados en HMMs el entrenamiento nos proporcionará, además de la matriz de transiciones A, el modelado acústico para el cual se emplea una mezcla de Gaussianas o GMMs.

$$b_j(\bar{o}) = \sum_{m=1}^{M} c_{jm} \cdot \mathrm{N}(\bar{o}; \mu_{jm}; \Sigma_{jm}) \qquad (2.14)$$

siendo:

M: número de Gaussianas de la mezcla.

c_{jm}: peso de la componente m-ésima.

$\mathrm{N}(\bar{o}; \mu_{jm}; \Sigma_{jm}) = \frac{1}{\sqrt{(2\cdot\pi)^n \cdot |\Sigma|}} \cdot e^{-\frac{1}{2}\cdot(\bar{o}-\mu_{jm})'\cdot(\Sigma_{jm})^{-1}\cdot(\bar{o}-\mu_{jm})}$: una Gaussiana multidimensional.

μ_{jm}: vector de medias.

Σ_{jm}: matriz de covarianza.

La etapa de entrenamiento se basa en el algoritmo Baum-Welch, mediante el cual se conseguirán estimar el vector de medias (μ) y la matriz de covarianzas (Σ).

El algoritmo Baum-Welch [15], es una generalización del algoritmo EM (Expectation-Maximization). Baum-Welch estima de forma iterativa los parámetros que definen las Gaussianas utilizadas para el modelado acústico (valores de μ y Σ), obteniendo al final del proceso los valores que mejor generarían o explicarían el conjunto de entrenamiento. El proceso iterativo seguido por el algoritmo parte de un segmentado inicial de las locuciones. Para este segmentado se calcularán las matrices de transición y los parámetros de las Gaussianas, con los cuales se realizará el segmentado, buscando maximizar la verosimilitud de las muestras de entrenamiento. Este proceso se repetirá mientras se siga maximizando la verosimilitud y se deberá llevar a cabo para todos los modelos/estados del sistema.

En lo que respecta al ámbito de este PFC, en el sistema híbrido HMM/SVM se reemplazan las GMMs por máquinas de vectores soporte, cuyo análisis se tratará en el Capítulo 3 de esta memoria.

2.3.3. Decodificación.

La decodificación en un reconocedor automático de habla basado en HMMs es llevada a cabo mediante el algoritmo de Viterbi. Para realizar este proceso, el decodificador se ayuda del diccionario de unidades básicas que se quieren reconocer (en este caso serán fonemas), además de una gramática. Ésta nos da información de las posibles transiciones entre las unidades contenidas en el diccionario.

En el reconocimiento de la locución se busca la secuencia de estados de mayor probabilidad para una secuencia dada de observaciones. El algoritmo de Viterbi busca el mejor camino a lo largo de una matriz (Trellis) en la que el eje vertical representa los estados de los HMMs y el horizontal son las tramas de habla (eje temporal). En la Figura 5, que se muestra más adelante, cada punto rojo en la cuadrícula representa la log-probabilidad de observar esa trama en ese instante concreto $\{b_j(\bar{o})\}$, mientras que las líneas verdes continuas que unen estos puntos son las log-probabilidades de transición $\{a_{ij}\}$.

Para un modelo 'M', denominamos $\phi_j(t)$ a la máxima verosimilitud acumulada de observar una secuencia determinada de estados, estando en el estado 'j' en el instante 't':

$$\phi_j(t) = \max_i \left\{ \phi_i(t-1) \cdot a_{ij} \right\} \cdot b_j(o_t) , \qquad (2.15)$$

con $\phi_1(1) = 1$ y $\phi_j(1) = a_{1j} \cdot b_j(o_t), 1 < j < N$.

A partir de esta variable se puede obtener la verosimilitud de una observación, dada una secuencia de estados, para un modelo concreto:

$$\hat{P}(o \mid M) = \phi_N(T) = \max_i \{\phi_i(T) \cdot a_{iN}\} \qquad (2.16)$$

Para evitar problemas de desbordamiento, se trabaja sobre el logaritmo de ϕ_j :

$$\psi_j(t) = \max_i \{\psi_i(t-1) + \log(a_{ij}) + \log(b_j(o_t))\} \qquad (2.17)$$

Para calcular la log-probabilidad acumulada por un camino del Trellis hasta el estado actual, debemos identificar los estados del instante anterior desde los que se puede llegar al estado actual e incrementarle la log-probabilidad asociada a la transición. Nos quedaremos con la log-probabilidad mayor y le sumaremos la log-probabilidad de la observación [16].

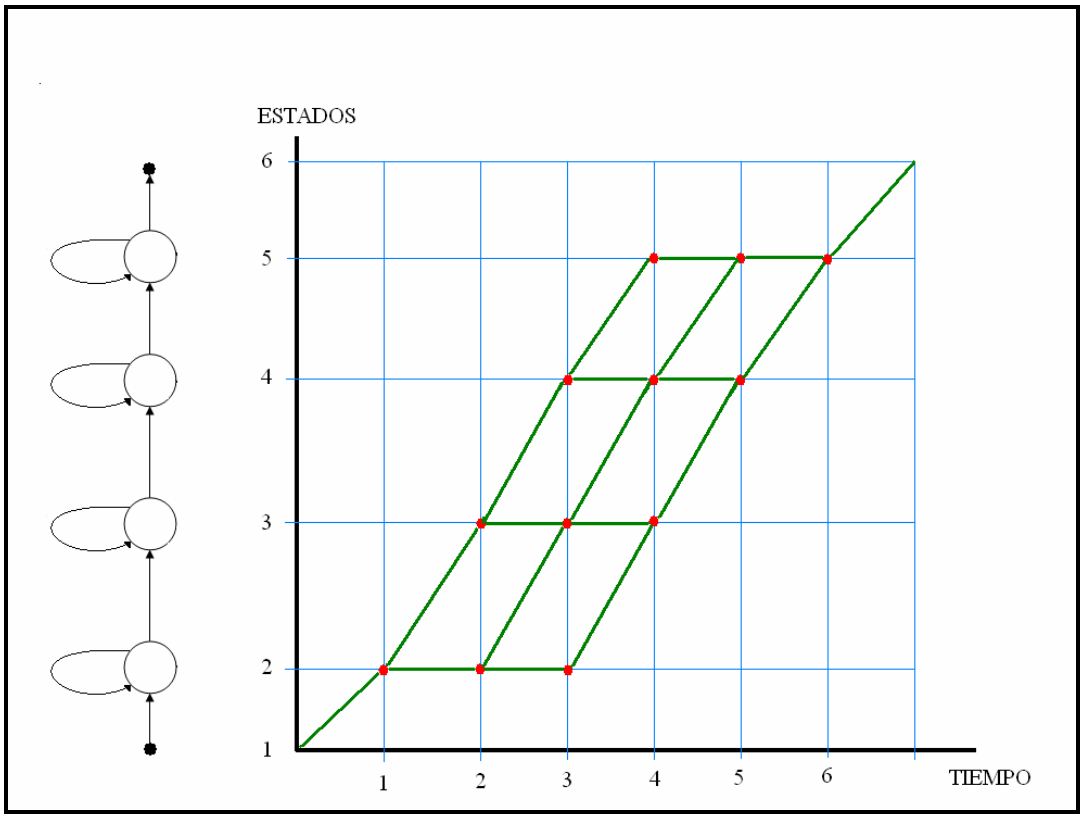


Figura 5. Algoritmo de Viterbi

Por último, como ampliación al algoritmo de Viterbi en el caso de reconocimiento de habla continua, se hablará del algoritmo 'Token Passing Model' [17], que sustituye al Viterbi en este tipo de reconocedores. El algoritmo 'Token Passing Model' es una modificación del algoritmo de Viterbi en el que se emplea un token que contiene información con la secuencia de palabras que se va decodificando.

Capítulo 3.
Reconocimiento Automático de Habla Híbrido HMM/SVM

En este capítulo se tratarán los sistemas de RAH híbridos más comunes basados en HMMs, los cuales han demostrado hasta el momento buenas propiedades en el reconocimiento de habla, pero tienen ciertas limitaciones que dificultan su aplicación en el mundo real. En primer lugar se hablará brevemente de los sistemas híbridos con redes neuronales. A continuación se describirá el funcionamiento de los sistemas híbridos con máquinas de vectores soporte. Este tipo de híbridos es el que se somete a estudio en este PFC. El capítulo finalizará hablando de los fundamentos matemáticos de las máquinas de vectores soporte, así como del papel que desempeñan los vectores soporte dentro de un híbrido HMM/SVM.

3.1. Introducción a los sistemas híbridos: HMM/ANN.

Las ANNs [18] tratan de imitar el cerebro humano y más en concreto la interconexión que existe entre las neuronas, tratando de conseguir mediante modelos matemáticos que las máquinas con esta tecnología respondan de la misma manera que lo hace el cerebro humano, dando respuestas generalizadas y robustas. Los primeros modelos de ANNs que se aplicaron al reconocimiento de habla fueron:

- TDNNs, o Redes de Retardo Temporal: convierten una secuencia temporal en una espacial mediante el desplazamiento de los vectores de entrada en cada una de las capas de la red, paliando de esta manera la dependencia temporal.
- RNNs, o Redes Neuronales Recurrentes: implementan lazos recursivos entre las capas de la red, desde la salida a la entrada.

Las ANNs pueden ser usadas en el reconocimiento de habla para la clasificación de fonemas o palabras, pero cuando se trata de reconocer habla continua sus resultados no son tan buenos como en los anteriores. Esto se debe a que las ANNs no son capaces de modelar adecuadamente secuencias de voz de distinta duración. Es aquí donde empiezan a cobrar importancia los híbridos basados en los HMMs.

Las dificultades de la aplicación directa de las ANNs en el RAH llevaron al desarrollo de los híbridos HMM/ANN. Estos híbridos utilizan MLPs (perceptrones multicapa) para la etapa de modelado acústico, en lugar de los GMMs. Los MLPs se componen de una capa de entrada, seguida de N capas intermedias ocultas y una capa de salida. Cada capa intermedia obtiene una combinación ponderada de sus entradas, para después pasar por una función no lineal, típicamente la que se muestra a continuación:

$$f(x) = \frac{1}{1+\exp(-x)} \tag{3.1}$$

Para las capas intermedias, esta función genera momentos de mayor orden que la entrada. Por otro, lado para la capa de salida la función anterior influirá en el modelado de la frontera de decisión. Para la capa de salida, una alternativa a la función (3.1), es la conocida como función softmax, que se muestra a continuación y donde K representa el número de unidades de esta capa de salida:

$$f(x_i) = \frac{\exp(x_i)}{\sum_{n=1}^{K} \exp(x_n)} \tag{3.2}$$

Dependiendo de la forma en que se use la ANN dentro de reconocedor HMM, se puede hablar de tres grupos posibles de híbridos HMM/ANN ([18],[19]):

- Para estimar las probabilidades a posteriori en cada uno de los estados del HMM. Se aprovechan la buenas cualidades de las ANNs como clasificadores y la posibilidad de proporcionar salidas “blandas” para obtener la probabilidad a

posteriori de cada una de las unidades acústicas consideradas, dada una observación.

- La ANN es entrenada para extraer las características relevantes de la voz, con el objetivo de transformar la parametrización de partida en una que sea más adecuada para ser modelada por las mezclas de Gaussianas en los HMM.

- Como cuantificadores vectoriales para HMM discretos: básicamente lo que pretenden es adaptar los valores reales de entrada a un conjunto limitado de valores con los que trabajará el HMM. De esta forma se hace que el HMM sólo se tenga que modelar para un conjunto finito de posibles observaciones.

De entre todos estos sistemas, los más usados son los primeros. Es en este tipo de sistemas híbridos sobre los que se trabaja en este Proyecto Fin de Carrera, pero empleando los híbridos de HMMs y SVMs, que se tratan en el siguiente punto.

3.2. Híbridos HMM/SVM.

Las SVMs ([20],[21]) fueron concebidas originalmente como clasificadores binarios, capaces de separar las muestras pertenecientes a dos clases mediante la determinación de una frontera de decisión, siempre y cuando las muestras pertenecientes a ambas clases fueran linealmente separables.

A continuación se presentarán los fundamentos matemáticos de las SVMs.

- *Muestras separables linealmente:*

En el planteamiento original se parte de la suposición de que las muestras son linealmente separables. Por lo tanto, se parte de un conjunto de muestras etiquetadas, al que llamaremos S y el cual emplearemos para el entrenamiento de la SVM:

$$S = \{(\overline{x}_1, y_1),...,(\overline{x}_n, y_n)\}, \text{ para } y_i \in \{-1,1\}, \overline{x}_i \in R^d \qquad (3.3)$$

donde $\overline{x}_i$ son las muestras de entrenamiento e y_i son las etiquetas que indica la clase a la que pertenece $\overline{x}_i$. Definiremos 'conjunto separable' como aquel en el que podemos clasificar los datos pertenecientes a ambas clases sin incurrir en error alguno. En este caso las muestras pertenecientes a ambas clases podrán ser separadas por un hiperplano de la forma:

$$\overline{w}^T \cdot \overline{x} + b = 0 \qquad (3.4)$$

donde $\overline{w}$ es un vector perpendicular al hiperplano buscado.

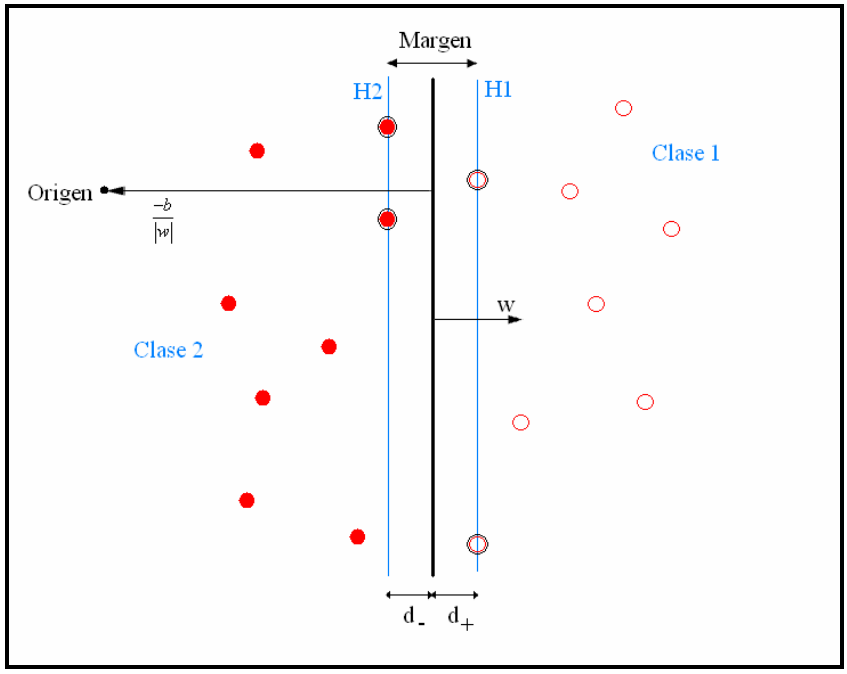


Figura 6. Hiperplano óptimo de dos clases linealmente separables

Por lo tanto, y dado que las muestras son separables, existirá al menos un hiperplano que cumpla:

$$\overline{w}^T \cdot \overline{x}_i^{(1)} + b \geq +1 \text{, para } y_i = +1 \qquad (3.5)$$

$$\overline{w}^T \cdot \overline{x}_i^{(-1)} + b \leq -1 \text{, para } y_i = -1 \qquad (3.6)$$

Estas inecuaciones se agrupan en una:

$$y_i \cdot (\bar{w}^T \cdot \bar{x}_i + b) - 1 \geq 0 \,,\ \forall i \qquad (3.7)$$

Definiremos el 'Hiperplano Óptimo de Decisión', como aquel que proporciona la mayor separación entre él y las muestras de entrenamiento, siempre que éstas hayan sido clasificadas sin errores. De las ecuaciones (3.5) y (3.6), obtendremos los hiperplanos siguientes:

H1: $\bar{w}^T \cdot \bar{x}_i^{(1)} + b = +1$, con distancia al origen $d_1 = \dfrac{|1-b|}{\|\bar{w}\|}$;

H2: $\bar{w}^T \cdot \bar{x}_i^{(-1)} + b = -1$, con distancia al origen $d_2 = \dfrac{|-1-b|}{\|\bar{w}\|}$;

El siguiente paso es maximizar la distancia entre el hiperplano óptimo y las muestras de la clase uno (d_+) y las de la clase dos (d_-). O lo que es lo mismo, hacer máxima la distancia entre H1 y (3.4) y de forma similar entre H2 y (3.4). De lo que obtendremos:

$$d_+ = d_- = \frac{1}{\|\bar{w}\|} \qquad (3.8)$$

El hiperplano buscado será aquel que maximice ($d_+ + d_-$), a lo que llamaremos 'margen'.

Partiendo del conocimiento de que H1 y H2 son paralelos podremos obtener el Hiperplano Óptino minimizando $\frac{1}{2} \cdot \|\bar{w}\|^2$, con la restricción de que cumpla la condición (3.7).

Para obtener la solución de este problema se debe recurrir a los multiplicadores de Lagrange ($\alpha_i \geq 0$), lo cual nos dará origen al 'Lagragiano':

$$L_p = \frac{1}{2}\|\bar{w}\|^2 - \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \left\{(\bar{w}^T \cdot \bar{x}_i + b) - 1\right\} \tag{3.9}$$

La expresión (3.9) se debe maximizar respecto a ' α ' y minimizar con respecto a 'w' y 'b'. Para ello se deben calcular las derivadas parciales del Lagragiano e igualarlas a cero:

$$\frac{\partial L_p}{\partial \bar{w}} = \bar{w} - \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \bar{x}_i = 0 \tag{3.10}$$

$$\frac{\partial L_p}{\partial b} = -\sum_{i=1}^{n} \alpha_i \cdot y_i = 0 \tag{3.11}$$

de donde se obtiene que:

$$\bar{w} = \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \bar{x}_i \tag{3.12}$$

$$\sum_{i=1}^{n} \alpha_i \cdot y_i = 0 \tag{3.13}$$

Sustituyendo en el Lagragiano, tendremos que:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \cdot \alpha_j \cdot y_i \cdot y_j \cdot (\bar{x}_i^T \cdot \bar{x}_j) \tag{3.14}$$

Maximizando la expresión anterior respecto a α y, teniendo en cuenta que $\sum_{i=1}^{n} \alpha_i \cdot y_i = 0$ y $\alpha_i \geq 0$, se llegaría a la solución:

$$f(x) = sign\left\{\sum_{i=1}^{n} \alpha_i \cdot y_i \cdot (\bar{x}_i^T \cdot \bar{x}) + b\right\} \tag{3.15}$$

donde $\alpha_i \neq 0$.

- *Muestras no separables linealmente:*

Para la resolución de este nuevo caso, se introducen unas nuevas variables $\xi_i \geq 0$, con i=1,..,n. La inclusión de estas nuevas variables hace que las ecuaciones que definen H1 y H2 queden de la siguiente manera:

$$\overline{w}^T \cdot \overline{x}_i^{(1)} + b \geq +1 - \xi_i, \text{ para } y_i = +1, \text{ donde } \xi_i \geq 0 \forall i \quad (3.15)$$

$$\overline{w}^T \cdot \overline{x}_i^{(-1)} + b \leq -1 + \xi_i, \text{ para } y_i = -1, \text{ donde } \xi_i \geq 0 \forall i \quad (3.16)$$

Como se ve, ξ_i permite un error en la clasificación. Estos nuevos factores hacen que la nueva función a minimizar quede de la siguiente manera:

$$\min\left\{\frac{1}{2}\cdot\left\|\overline{w}\right\|^2 + C\cdot\sum_{i=0}^{n}(\xi_i)^{\sigma}\right\} \quad (3.17)$$

donde C es un factor de penalización de los errores cometidos durante el entrenamiento y σ cualquier entero positivo. Si toma el valor 1, el Lagragiano quedaría:

$$L_p = \frac{1}{2}\cdot\left\|\overline{w}\right\|^2 + C\cdot\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\mu_i\cdot\xi_i - \sum_{i=1}^{n}\alpha_i\cdot\left\{y_i\cdot(\overline{w}^T\cdot\overline{x}_i + b) - 1 + \xi_i\right\}; \quad (3.18)$$

Aplicando las derivadas parciales e igualando a cero, tal como se hizo para el caso lineal, se obtiene:

$$L_D = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\cdot\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\cdot\alpha_j\cdot y_i\cdot y_j\cdot(\overline{x}_i^T\cdot\overline{x}_j) \quad (3.19)$$

Al igual que en el punto anterior, maximizando esta expresión respecto a α_i y teniendo en cuenta las condiciones $0 \leq \alpha_i \leq C$, $\sum_{i=1}^{n}\alpha_i\cdot y_i = 0$, llegaríamos a la solución buscada.

Puede suceder que los datos no sean linealmente separables, por lo que se puede optar por fronteras de decisión no lineales. Dado que la solución de la SVM queda expresada en todo momento como productos escalares de las muestras de entrada, se pueden sustituir dichos productos escalares por lo que se denomina "kernel". El kernel induce implícitamente una transformación no lineal sobre los datos de entrada, obteniéndose fronteras de decisión no lineales. Algunos ejemplos de kernel son:

- Lineal: $k(\overline{x}_i, \overline{x}_j) = \overline{x}_i^T \cdot \overline{x}_j$.
- Polinómico homogéneo: $k(\overline{x}_i, \overline{x}_j) = (\overline{x}_i^T \cdot \overline{x}_j)^a$.
- Polinómico inhomogéneo: $k(\overline{x}_i, \overline{x}_j) = (\overline{x}_i^T \cdot \overline{x}_j + c)^a$, donde $a \in \mathrm{N}$ y $c \geq 0$.
- Sigmoidal: $k(\overline{x}_i, \overline{x}_j) = \tanh(\upsilon(\overline{x}_i^T \cdot \overline{x}_j) + \vartheta)$.
- RBF: $k(\overline{x}_i, \overline{x}_j) = \exp\left(-\frac{\left\|\overline{x}_i - \overline{x}_j\right\|^2}{2 \cdot \sigma^2}\right)$.
- Exponencial: $k(\overline{x}_i, \overline{x}_j) = \exp\left(-\frac{\left\|\overline{x}_i - \overline{x}_j\right\|}{2 \cdot \sigma^2}\right)$

El más común es el RBF, que es el que ha sido empleado en la SVM de este Proyecto. Para hacer uso del kernel únicamente se ha de sustituir $\overline{x}_i^T \cdot \overline{x}_j$, por $K(\overline{x}_i, \overline{x}_j)$ en la fase de entrenamiento, con lo que se tendría:

$$\max_{\alpha} \left\{ L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \cdot \sum_{i,j=1}^{n} \alpha_i \cdot \alpha_j \cdot y_i \cdot y_j \cdot K(\overline{x}_i, \overline{x}_j) \right\} \qquad (3.20)$$

donde se debe cumplir que $\sum_i \alpha_i \cdot y_i = 0$ y que $0 \leq \alpha_i \leq C$.

Finalmente obtendríamos que la solución para el caso de muestras linealmente no separables es:

$$f(x) = sign\left\{\sum_{i=1}^{n} \alpha_i \cdot y_i \cdot K(\overline{x}_i, \overline{x}_j) + b\right\} \qquad (3.21)$$

- *Extensión a un problema de clasificación con más de dos clases:*

Como se indicaba al comienzo de este capítulo, las SVMs originalmente nacieron como máquinas de clasificación binarias. Esto quiere decir que sólo son capaces de clasificar las muestras en dos clases, lo cual es una gran barrera, ya que en la gran mayoría de problemas de RAH se consideran múltiples clases. Particularizando sobre este PFC, tenemos 33 clases. El método de resolución propuesto para el caso multiclase [22] consiste en obtener varios clasificadores, en concreto uno por cada pareja de clases que compongan el problema, por lo que si N es el número total de clases se necesitan $\frac{N \cdot (N-1)}{2}$ clasificadores distintos. A esta aproximación se la denomina 1 contra 1. Las salidas de todos estos clasificadores se deben combinar adecuadamente para obtener las probabilidades a posteriori de cada una de las clases consideradas.

La forma en que se usarán las SVMs en el sistema híbrido HMM/SVM consiste en sustituir las GMMs de los sistemas tradicionales basados en HMMs por las SVMs, de modo que serán éstas las encargadas de proporcionar las probabilidades a posteriori de cada unidad acústica. Dado que estamos ante una SVM multiclase, donde se ha empleado una clasificación 1 contra 1, podemos calcular las probabilidades a posteriori de la SVM binaria (i,j) (τ_{ij} = p(y = i|y = i ó j,x)) aplicando una función sigmoidal sobre la salida "blanda" de dicha SVM binaria:

$$\tau_{ij} \approx \frac{1}{1 + e^{A \cdot f(x) + B}} \qquad (3.22)$$

Los valores A y B son estimados a partir de las salidas de la SVM binaria, obtenidas a partir de un proceso de validación cruzada de una parte de la base de datos.

Una vez calculados los valores τ_{ij} para las $\frac{N \cdot (N-1)}{2}$ SVMs binarias se podrán obtener las probabilidades a posteriori (p_i) de las N clases que forman el diccionario de la base de datos:

$$p_i = p(y = i \mid x) \text{ , i = 1,…, N} \tag{3.23}$$

Para obtener las probabilidades p_i se emplea una variante del algoritmo de Refregier y Vallet [23], en el que se minimiza el siguiente funcional:

$$\min_{p} \sum_{i=1}^{k} \sum_{j:j \neq i} (\tau_{ji} \cdot p_i - \tau_{ij} \cdot p_j)^2 \tag{3.24}$$

donde $\sum_{i=1}^{k} p_i = 1$, $p_i \geq 0, \forall i$. Resolviendo dicho funcional de forma iterativa se obtendrán las probabilidades deseadas.

3.3. Interpretación de los Vectores Soporte.

Se denominan vectores soporte a las muestras de entrenamiento que definen o delimitan el hiperplano óptimo de decisión; por lo tanto, la decisión de la clase a la que se asigna una nueva muestra o la probabilidad a posteriori de dicha clase dependerá sólo de estos SVs. Si observamos la Figura 8, representada en el punto 3.2, los vectores soporte son aquellas muestras del conjunto de entrenamiento que se sitúan sobre las fronteras H1 y H2 y que están contenidas en un círculo negro, así como aquellas que están dentro del margen (bien clasificadas) y las que están mal clasificadas.

Los SVs tienen $\alpha_i \neq 0$: esto implica que el hiperplano óptimo de decisión y la salida del clasificador (ecuación 3.21), se puede calcular únicamente en base a las muestras elegidas como vectores soporte.

El objetivo de este PFC es darle un sentido físico a los SVs, buscando si es posible establecer algún tipo de relación entre estos SVs y las fronteras de cada uno de los fonemas considerados, o bien si es posible localizarlos en alguna zona concreta del fonema. Hay que tener en cuenta que el entrenamiento de la SVM es supervisado, es decir, las muestras de entrenamiento consisten en pares vector-etiqueta, lo que hace posible determinar la clase a la que pertenece cada SV.

Una vez que la máquina de vectores soporte ha sido entrenada, tarea para la cual se emplea el software LibSVM [2], podemos extraer el conjunto de los SVs que determinan la frontera de decisión, para cada una de las 33 clases que se corresponden con cada una de las unidades acústicas consideradas. Con los SVs que hemos obtenido se tratará de determinar si se puede inferir alguna regla que haga que la máquina de vectores soporte elija estos vectores y no al resto de vectores que conforman la BBDD y que no son catalogados como SVs.

El primero de los pasos que se darán para realizar este análisis es la implementación de una herramienta gráfica o GUI que nos facilite el análisis visual del emplazamiento de los SVs. El segundo paso será una comparación estadística en la ubicación de los SVs con respecto a las transiciones que delimitan los fonemas a los que representan.

3.3.1. Herramienta Gráfica.

Para la implementación de la herramienta gráfica se decidió recurrir al software Matlab. Este software, además de realizar cálculos matemáticos, facilita la implementación de entornos gráficos que hacen posible la interactividad del usuario con los datos que maneja el programa. Por sus características, este software permite realizar cálculos con vectores y matrices de grandes dimensiones, lo cual resulta bastante adecuado para manejar el gran volumen de datos con el que debemos trabajar.

Los detalles sobre el análisis gráfico de esta herramienta se presentan con más detalle en el capítulo 4 de este Proyecto.

3.3.2. Comparación.

Para realizar el estudio en la ubicación de los SVs es preciso tener una referencia previa. Dado que cada SV interviene en la definición de la frontera de decisión de una sola clase y cada clase equivale a un fonema, las referencias que buscamos deben ser las transiciones entre los distintos fonemas. Como referencia se tomará el segmentado realizado por el RAH híbrido HMM/SVM y un segmentado manual. Los resultados sobre este estudio se detallan en el capítulo 5 de este Proyecto.

Capítulo 4.
Herramienta Gráfica para la Interpretación de los Vectores Soporte

Para llevar a cabo el estudio de la situación de los SVs en las locuciones y su relación con las transiciones entre fonemas, se ha diseñado una herramienta gráfica que nos permite su estudio cualitativo (de forma visual) directamente sobre la señal de voz. En este capítulo se realizará una descripción de la herramienta gráfica desarrollada. Se hablará sobre el tipo de datos que toma como entrada y las diferentes representaciones que realiza sobre ellos.

4.1. Base de Datos empleada.

La base de datos que se ha empleado en este PFC es la base de datos en castellano SpeechDat II FDB – 4000 [24], con 160.000 locuciones. Estas locuciones pueden contener dígitos aislados o conectados, fechas, deletreo de palabras, cantidades monetarias y direcciones, entre otras. La grabación de estos ficheros de la base de datos fue realizada por múltiples locutores, con una amplia variabilidad de pronunciación y en distintos ambientes sin ruido. Los ficheros de esta base de datos están muestreados a una tasa de 8 KHz y 8 bits/muestra con cuantificación según Ley-A.

4.2. Entrenamiento de la SVM.

La SVM multiclase de este experimento deberá clasificar las muestras de la base de datos como una de las 33 clases consideradas. Cada una de estas 33 clases se corresponde con un determinado fonema de la siguiente lista:

1	*B*	**7**	*T*	**13**	*g*	**19**	*m*	**25**	*S*	**31**	*w*
2	*D*	**8**	*a*	**14**	*i*	**20**	*n*	**26**	*Sil*	**32**	*x*
3	*G*	**9**	*b*	**15**	*j*	**21**	*o*	**27**	*Sp*	**33**	*z*
4	*J*	**10**	*d*	**16**	*jj*	**22**	*p*	**28**	*T*		
5	*L*	**11**	*e*	**17**	*k*	**23**	*r*	**29**	*tS*		
6	*N*	**12**	*f*	**18**	*l*	**24**	*rr*	**30**	*U*		

Tabla 1. Relación Clase/Fonema

Para entrenar la SVM se escogió un grupo de 16.000 ficheros, que componen un 10% del total de la base de datos completa. En estos ficheros de entrenamiento están representados todos los tipos de locuciones que componen la base de datos. El limitar el tamaño de la base de datos empleada para entrenar la SVM es debido al elevado coste computacional que supone dicho entrenamiento, el cual aumenta considerablemente a medida que lo hace el tamaño de la base de datos. Por un lado tenemos que en este grupo de ficheros, el número de fonemas pertenecientes a cada una de las 33 clases no se reparte de manera uniforme. Y por el otro sabemos que en las SVM puede ser conveniente que el número de elementos de cada una de sus clases sí sea el mismo. Por lo tanto un primer paso es conocer la probabilidad de aparición de cada fonema y realizar una preselección sobre ellos. Una vez compensado el número de muestras por cada clase de la SVM, el siguiente paso es escoger los parámetros óptimos de ponderación (‘C’) y anchura del kernel RBF gamma (‘G’) de la SVM, que maximicen la tasa de aciertos de la máquina. Para ello se realiza un proceso de validación cruzada sobre el conjunto de muestras de entrenamiento, variando en cada ensayo los valores de estos dos parámetros ($C = [2^4 - 2^{15}]$ y $G = [2^{-3} - 2^{-9}]$. Una vez seleccionados los valores óptimos se procede a entrenar la máquina usando dichos valores de ‘C’ (2^6) y ‘G’ (2^{-5}). Por último sólo nos quedaría realizar el ´test’ a nivel de trama sobre un conjunto de 1.000 ficheros que únicamente contenían dígitos y que evidentemente no coincide con la de entrenamiento. Se obtienen unos resultados del 41,68%.

4.3. Localización de los SVs.

Tras la ejecución del entrenamiento de la SVM se obtiene un fichero ‘.model’ con la siguiente cabecera:

```
svm_type c_svc
kernel_type rbf
gamma 0.03125
nr_class 33
total_sv 30090
rho 2.83678 -2.77028 3.40475 1.97345 -0.656939 -0.198368 …
label 10 11 12 13 14 15 16 17 18 19 1 20 21 22 23 24 25 26 27 28 29 2 30 31 32 33 3 4 5
6 7 8 9
nr_sv 958 887 913 948 842 923 939 927 920 960 938 981 960 907 949 975 879 1010 942 951
808 1009 826 780 784 893 999 855 934 849 962 768 914
```

Figura 7. Cabecera fichero '.model'

donde podemos ver los distintos parámetros con los que se configuró la SVM. Los valores que nos interesan para poder extraer todos los SVs incluidos en el modelo serían:

- Número de clases: 33, correspondiente al número de fonemas de nuestro diccionario (Tabla 1).
- Número toral de vectores soporte: 30090.
- Etiquetas: identifica cada clase (o fonema en este caso) de la máquina (Tabla 1).
- Número de SVs: indica el número de SVs por cada clase, según el orden en el que aparecen los identificadores de las etiquetas.
- Por último y como continuación a esta cabecera vendrían todos los SVs.

Usando Matlab se extraen los SVs del fichero, etiquetando cada uno con el identificador de la clase a la que pertenece. El siguiente paso es la búsqueda de cada SV dentro de la base de datos de entrenamiento. Para ello se recorre la base de datos parametrizada, buscando para cada SV la trama de la BBDD que tenga una menor distancia euclídea con el SV. Esta minimización de la distancia euclídea ha sido necesaria debido a que los valores en el modelo y en la parametrización aparecen con distinta precisión en el número de decimales. Una vez hallado el vector, para poder ubicarlo y recuperarlo de una forma rápida, se marca con el número del fichero al que pertenece, la línea dentro del fichero y la clase de la SVM a la que se asoció.

Para poder realizar el estudio pretendido, no sólo necesitamos la posición de cada SV dentro de su fichero, sino su posición relativa en el fonema al que representa. Por lo tanto, el siguiente paso es encontrar los instantes temporales donde se producen las transiciones entre fonemas de cada fichero.

Un primer método para realizar esta búsqueda de transiciones entre fonemas consistió en el segmentado manual de parte de la base de datos. Dado que realizar este proceso de forma manual, basándose únicamente en la percepción del oído humano y la visualización de la forma de onda y el espectrograma de la señal, es un proceso muy costoso, se tomó la decisión de segmentar sólo parte de la base de datos. Para determinar el tamaño se tomó como criterio el elegir un número de ficheros tal que contuvieran una cantidad similar de fonemas de cada clase o que al menos, contuviera un número suficiente de ejemplos del fonema menos presente. El número de ficheros segmentados manualmente fue de 254, dado que se fijó un número mínimo de fonemas por clase de 50. Estos 254 ficheros representan un 2% de la base de datos total, los cuales agrupan un total de 1808 SV (6% del total). La distribución de los SVs por clase se muestra en la Tabla 2.

/B/	/D/	/G/	/J/	/L/	/N/	/T/	/a/
79	50	51	53	51	50	59	58
/b/	/d/	/e/	/f/	/g/	/i/	/j/	/jj/
68	50	51	64	51	61	61	53
/k/	/l/	/m/	/n/	/o/	/p/	/r/	/rr/
50	50	53	51	50	51	56	50
/s/	/sil/	/sp/	/t/	/tS/	/u/	/w/	/x/
50	57	58	50	56	50	57	57
/z/							TOTAL
52							1808

Tabla 2. Distribución número de SV por clase

El segundo método empleado se basa en el segmentado de los ficheros realizado por el híbrido HMM/SVM. Durante el proceso de reconocimiento, el híbrido marca para cada instante temporal qué muestra ha reconocido. Procesando adecuadamente estos ficheros se pueden obtener las transiciones entre los fonemas, tal y como son detectados por el híbrido.

De los resultados obtenidos y la comparación entre los distintos métodos se hablará en el siguiente capítulo con mayor detenimiento.

4.4. Representaciones.

En este apartado se expondrán las distintas representaciones que permite realizar la herramienta desarrollada, dejando la explicación sobre su uso y los detalles sobre su programación para los Anexos de este Proyecto Fin de Carrera.

La Figura 8 muestra el interfaz de la herramienta gráfica diseñada, donde seleccionando diversas combinaciones de las opciones que ofrece, se pueden mostrar distintas representaciones de los ficheros/locuciones que componen la BBDD y sus vectores soporte asociados.

Como opción adicional se incluye el poder escuchar el fichero seleccionado. Se decidió la inclusión de esta opción en la herramienta, pese a no ser un elemento visual, para ayudar en la interpretación de las figuras representadas y su asociación con el contenido vocal del fichero.

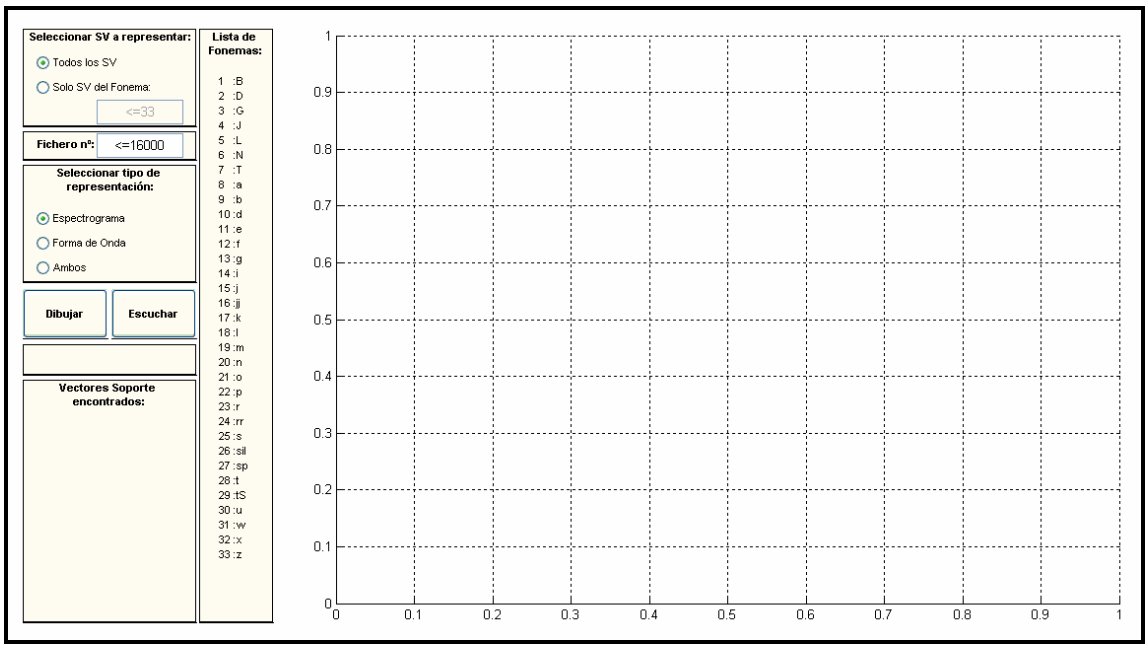


Figura 8. Interfaz de la Herramienta Gráfica

4.4.1. Señal de voz original en el dominio del tiempo.

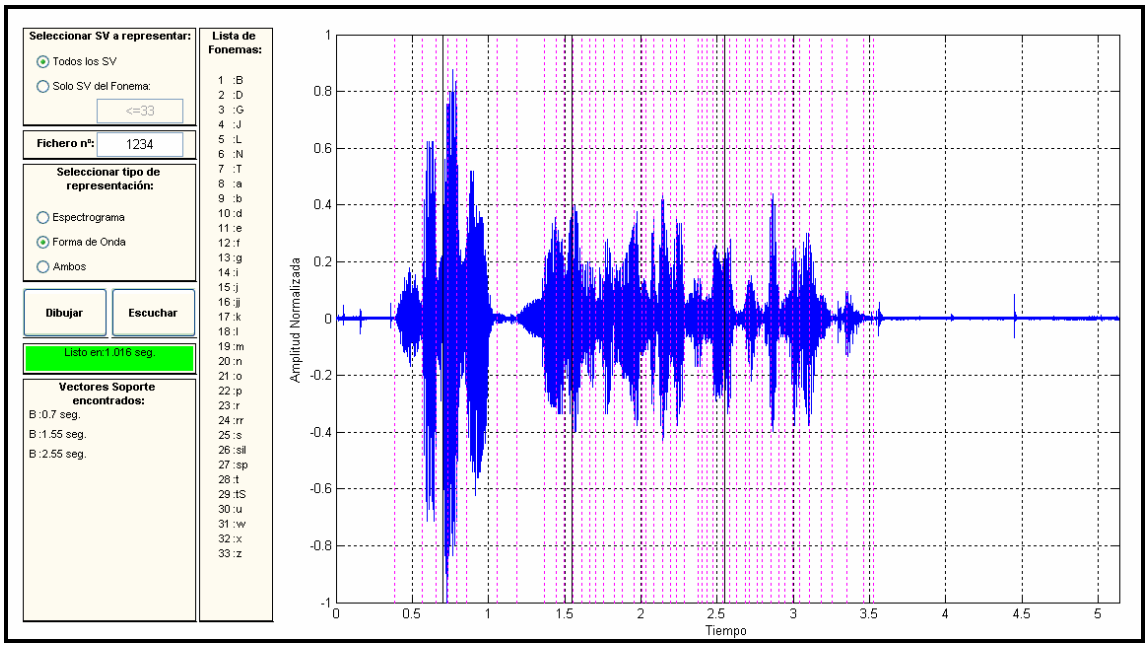


Figura 9. Representación de Señal en el Tiempo

Esta opción nos muestra la representación en el dominio del tiempo del fichero de la base de datos seleccionado. La escala temporal se muestra en segundos, mientras que la amplitud se encuentra normalizada a ±1. Se incluyen unas marcas verticales que indican las transiciones entre fonemas (líneas rosas de puntos), para facilitar la localización de los vectores soporte respecto a los distintos fonemas. Estas transiciones son las obtenidas mediante el segmentado manual.

Por la forma de onda podemos distinguir si un determinado fonema es una vocal o una consonante. Las vocales tienen una forma de onda armónica, ya que se producen por una vibración de las cuerdas vocales, mientras que los fonemas tienen una apariencia ruidosa, debido a que su articulación se produce interponiendo los distintos elementos del tracto vocal (lengua, dientes, labios) al flujo de aire desde los pulmones hacia el exterior.

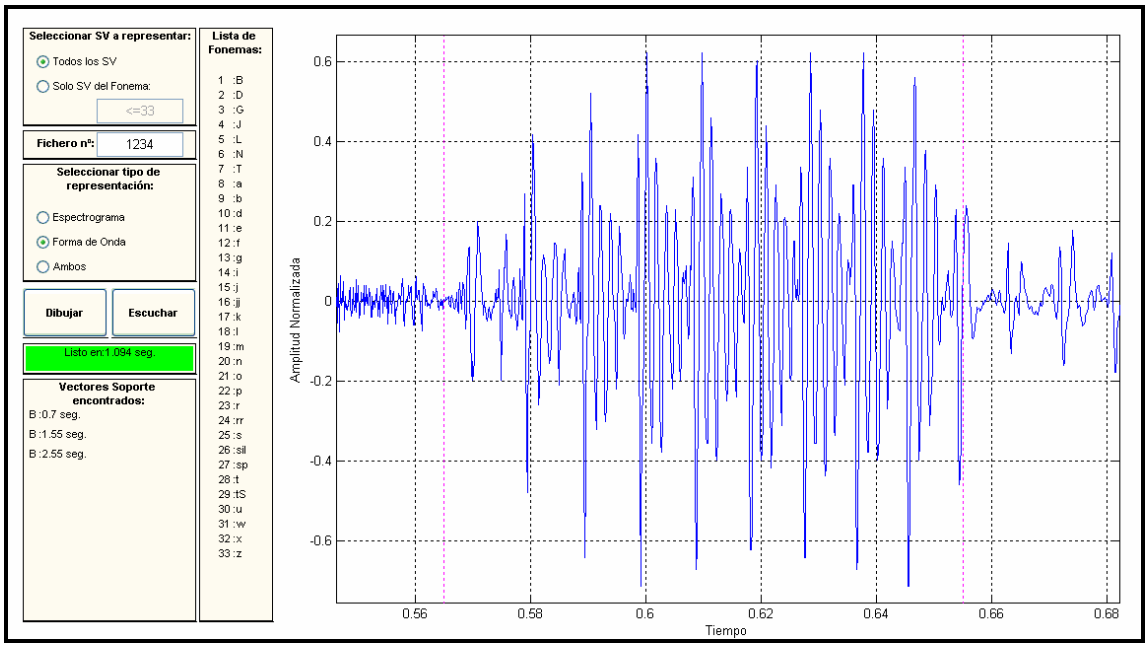


Figura 10. Representación fonema /a/

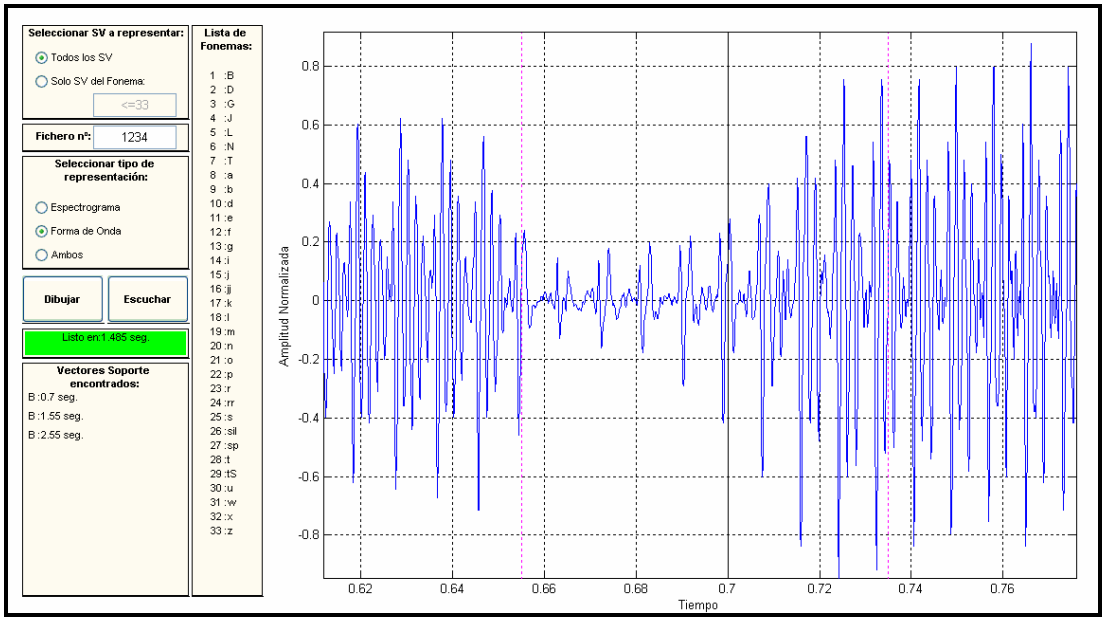


Figura 11. Representación fonema /B/

En la primera de las dos imágenes anteriores se puede apreciar la representación de la forma de onda del fonema /a/, así como su duración y su amplitud normalizada. En la parte central de la segunda figura se observa un fonema /B/, donde además en el instante 0.7 segundos se ha ubicado uno de los SV seleccionados por la máquina. Los SV se representan en la GUI como líneas negras continuas.

4.4.2. Espectro de la señal de voz original.

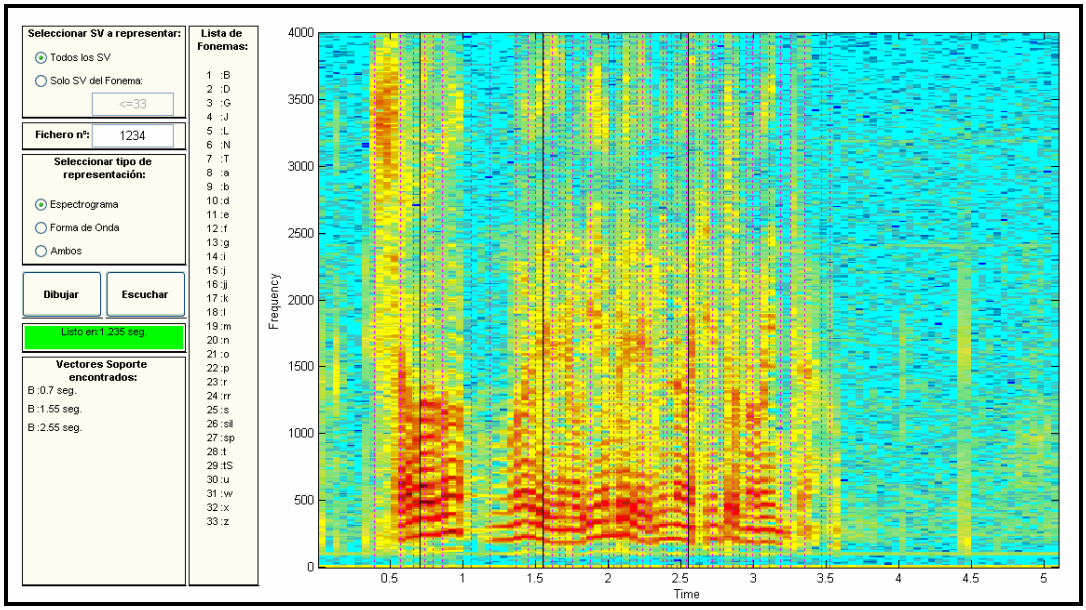


Figura 12. Representación Señal en Frecuencia

La representación del espectrograma de una señal nos da información sobre la frecuencia fundamental, sus armónicos y los distintos formantes de la voz del locutor. El espectrograma que muestra esta GUI es de banda estrecha, aunque modificando el código fuente (de fácil acceso en Matlab) se puede obtener un espectrograma de banda ancha.

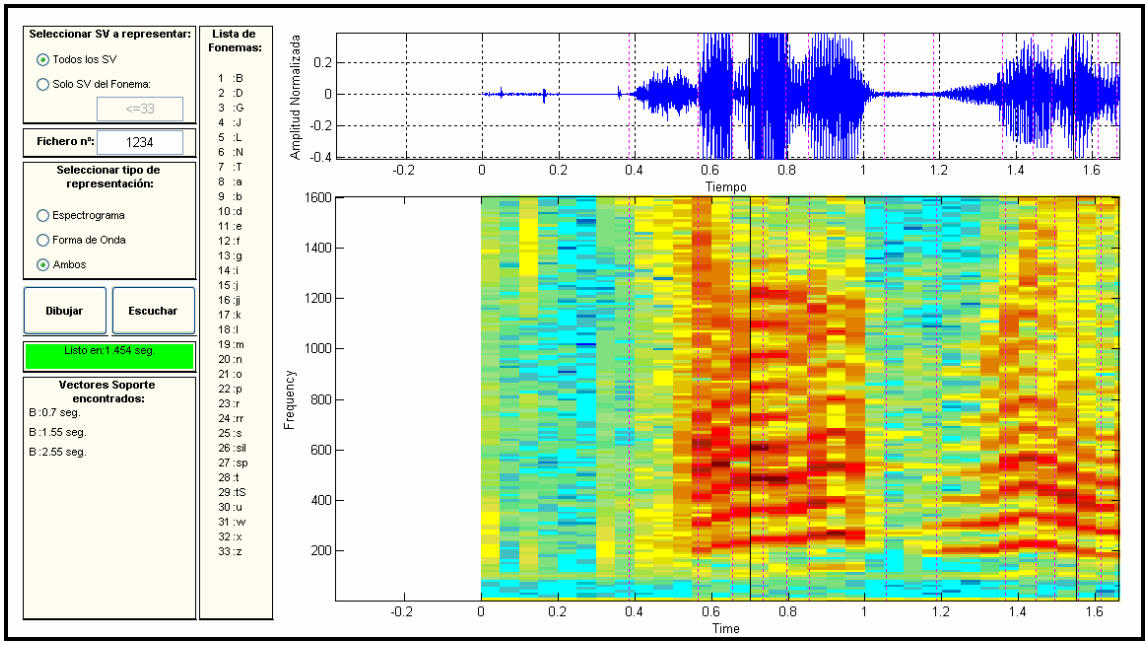


Figura 13. Representación combinada Tiempo-Frecuencia

Como complemento, la GUI añade la posibilidad de visualizar ambas representaciones (temporal y frecuencial) de forma simultánea, lo que nos permite combinar las propiedades de las dos vistas.

4.4.3. Todos los SVs en un fichero.

En los dos apartados anteriores se ha hablado de los tipos de representaciones que permite la GUI. En estas representaciones, por defecto, aparecerán todos los SV que hayan sido encontrados en el fichero seleccionado, independientemente de su cantidad y clase a la que pertenezcan.

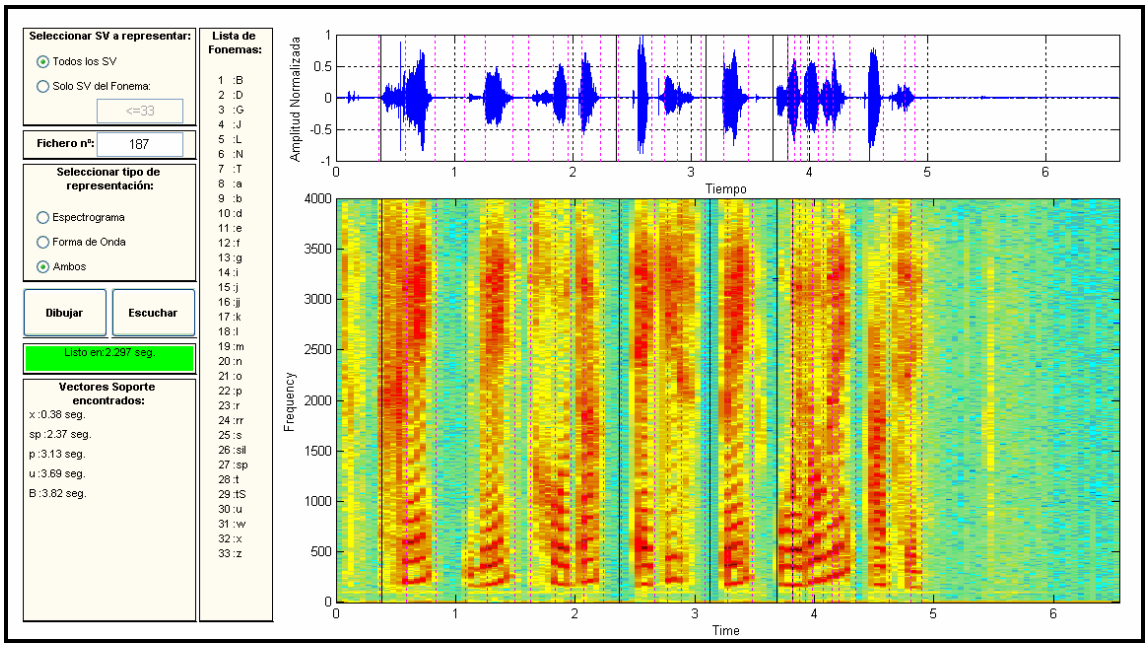


Figura 14. Señal con todos sus SV

Analizar en qué ficheros recae un mayor número de SV puede ser de utilidad para analizar cómo influye el ruido o las características concretas de los locutores sobre la selección de estos vectores soporte.

4.4.4. Todos los SVs de una clase en un fichero.

Por último, puede resultar conveniente que la herramienta únicamente nos represente los SVs ubicados en el fichero seleccionado y que pertenezcan a una determinada clase. Como se ve en el ejemplo de la Figura 15, de los cinco vectores soporte que fueron encontrados en este fichero y que se mostraban en la figura del apartado 4.4.3, se ha configurado la herramienta para que sólo presente el correspondiente al fonema /u/.

Este tipo de representación será de utilidad cuando se esté haciendo un estudio o seguimiento de los SV de una determinada clase, ya que nos eliminará los elementos representados en pantalla que no sean de utilidad en ese momento.

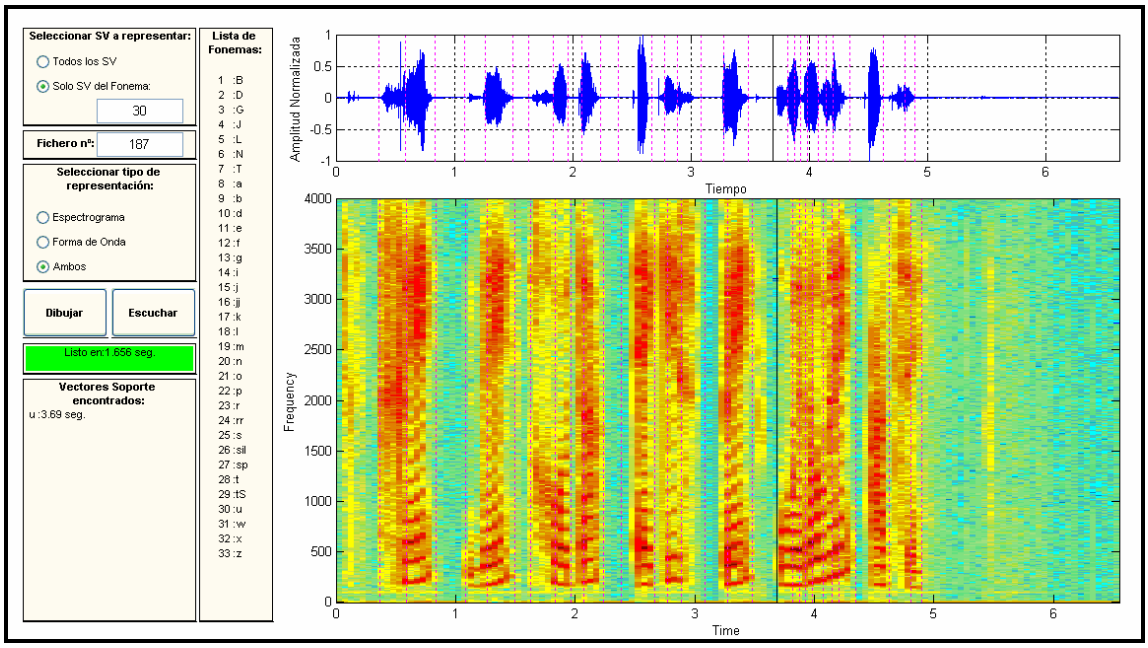


Figura 15. Filtrando por SV del fonema /u/

Capítulo 5.
Estudio Estadístico de la Localización de los SVs

En este capítulo se presentan y analizan los resultados obtenidos sobre la ubicación de los SVs con respecto a los dos tipos de marcado que se han realizado de las transiciones entre fonemas. Con el fin de poder obtener unas conclusiones más objetivas, los resultados que se muestran en este capítulo están obtenidos en base al mismo número de ficheros, ya que, como se indicaba en el capítulo anterior, el coste del segmentado manual impidió realizar el estudio con un mayor número de locuciones de la base de datos.

El estudio estadístico se ha basado en la elaboración de histogramas, en los cuales se puede apreciar la distribución de los SVs sobre la duración normalizada de los fonemas. Además se ha añadido a cada gráfica el valor de la media y varianza de cada conjunto de datos representado.

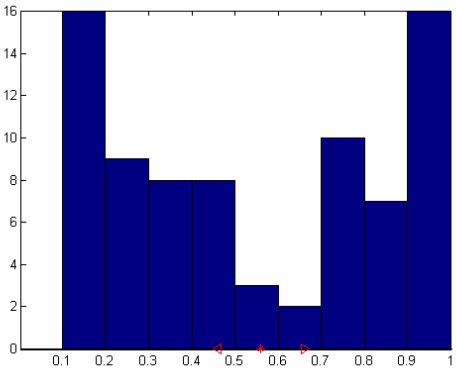
5.1. Comparación con segmentado manual.

Para la realización de este segmentado manual fue necesario el uso del programa 'Cool Edit'. Esta herramienta nos permite realizar un seguimiento sobre el espectrograma o la forma de onda de la señal de forma simultánea a la reproducción acústica del fichero. De esta forma el segmentado no se basó únicamente en la percepción auditiva sino también en los cambios de la señal (espectrograma o forma de onda).

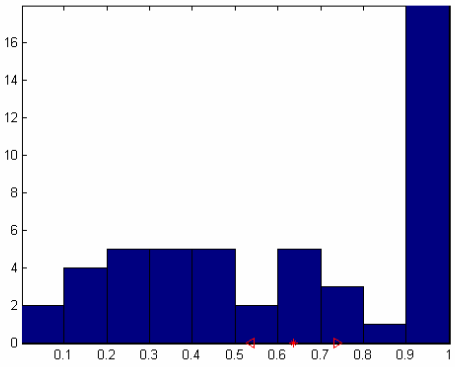
A continuación se mostrarán los histogramas obtenidos para la posición de los SVs en cada fonema y finalmente se presentará un cuadro resumen con los valores de la media y varianza de cada uno de ellos. Los histogramas representan en el eje horizontal

la duración normalizada de los fonemas, siendo 0 el comienzo del fonema y 1 el final del mismo.

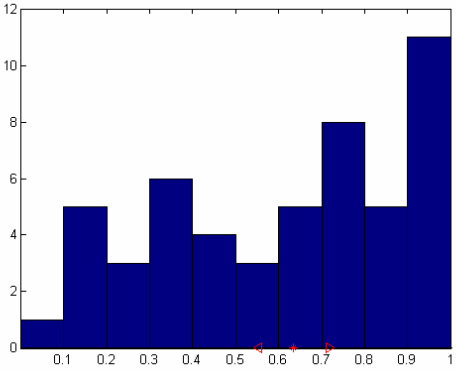
Fonema: /B/



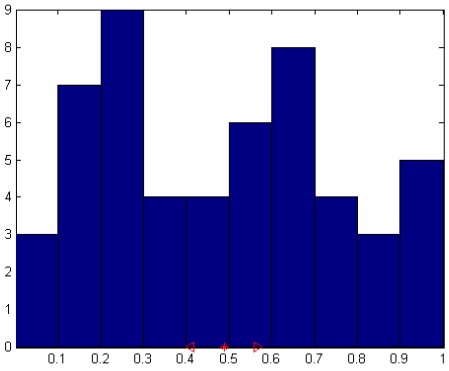
Fonema: /D/



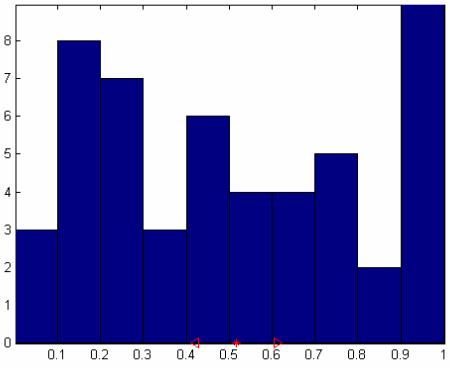
Fonema: /G/



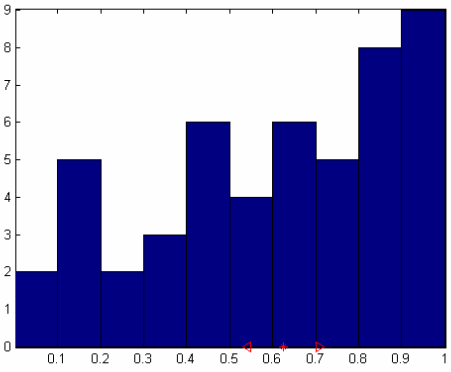
Fonema: /J/



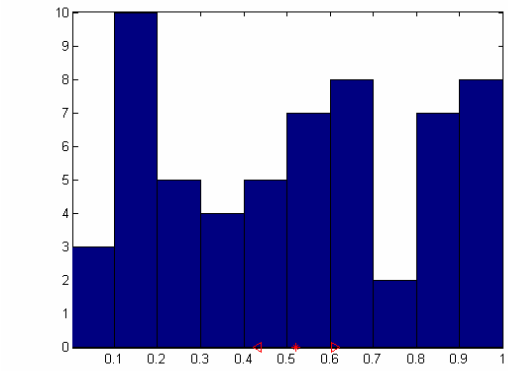
Fonema: /L/



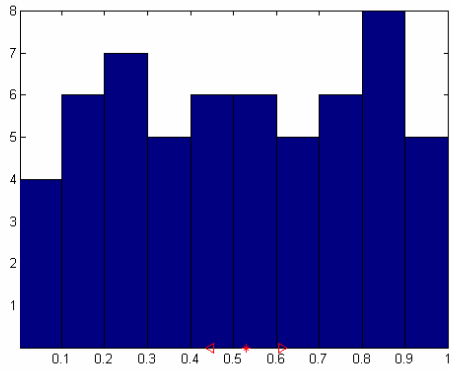
Fonema: /N/



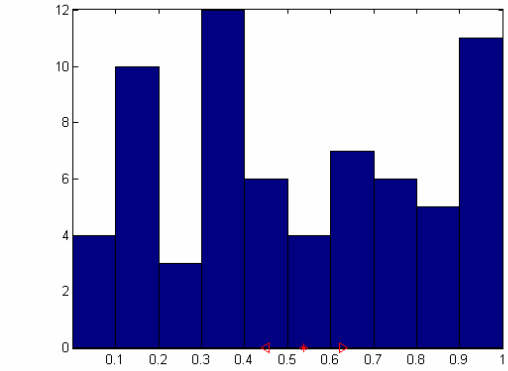
Fonema: /T/



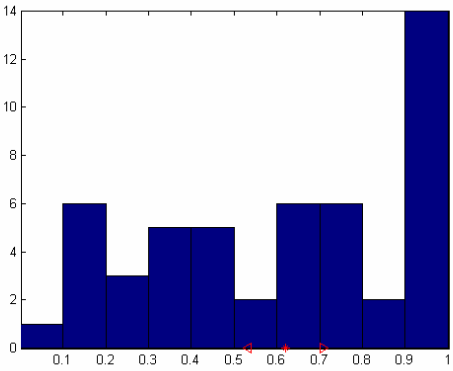
Fonema: /a/



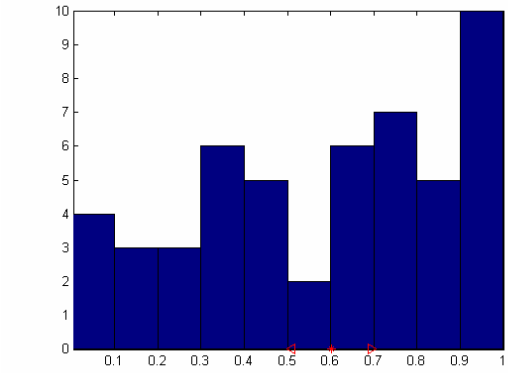
Fonema: /b/



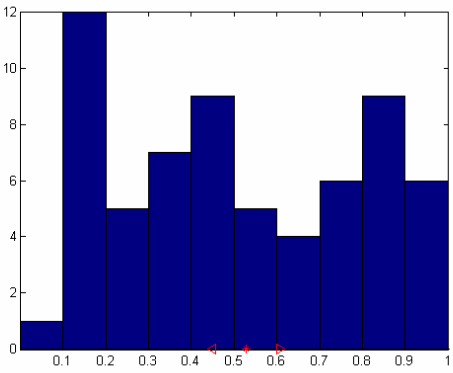
Fonema: /d/



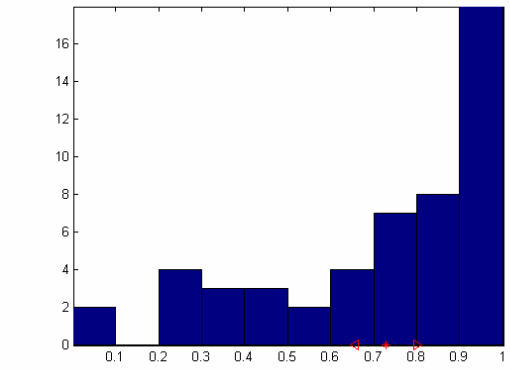
Fonema: /e/



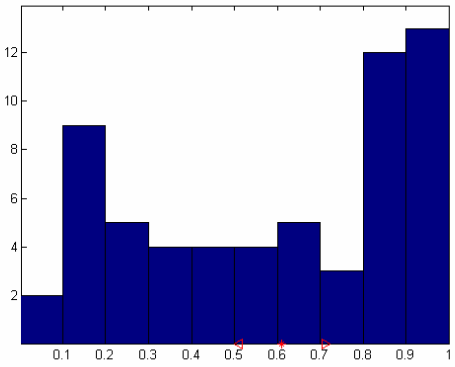
Fonema: /f/



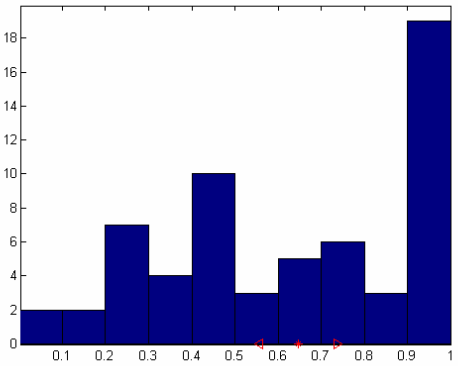
Fonema: /g/



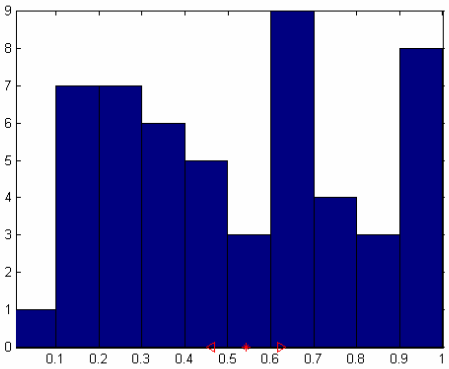
Fonema: /i/



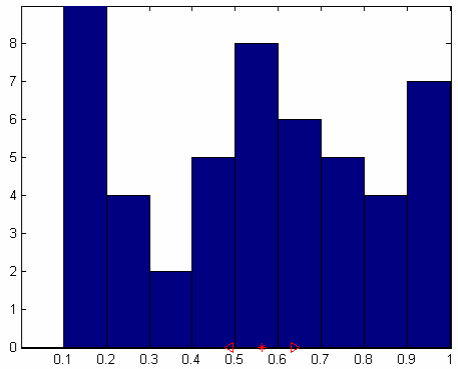
Fonema: /j/



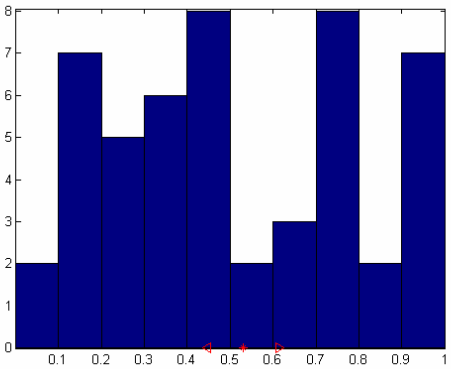
Fonema: /jj/



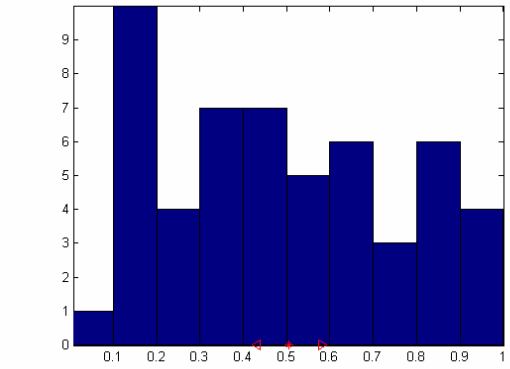
Fonema: /k/



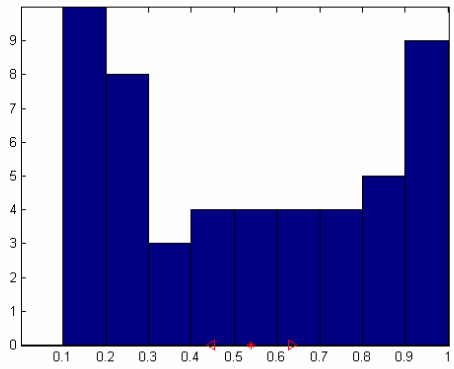
Fonema: /l/



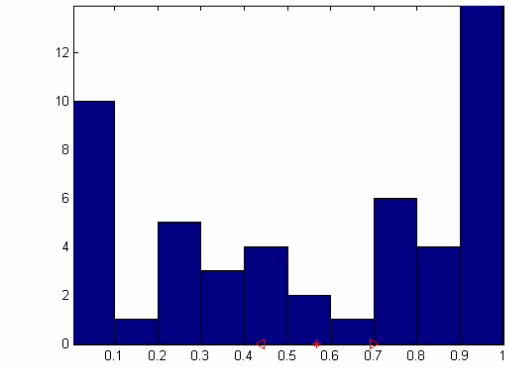
Fonema: /m/



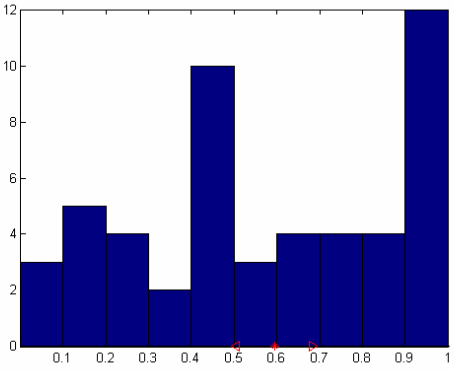
Fonema: /n/



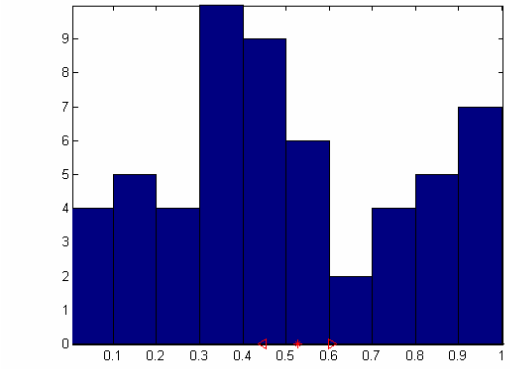
Fonema: /o/



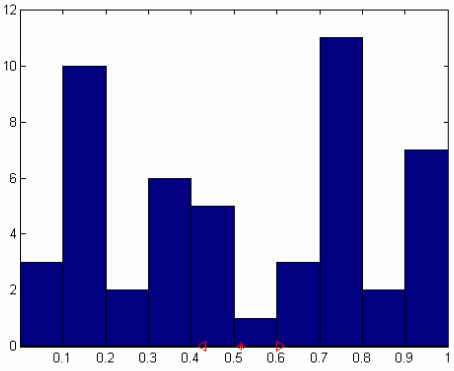
Fonema: /p/



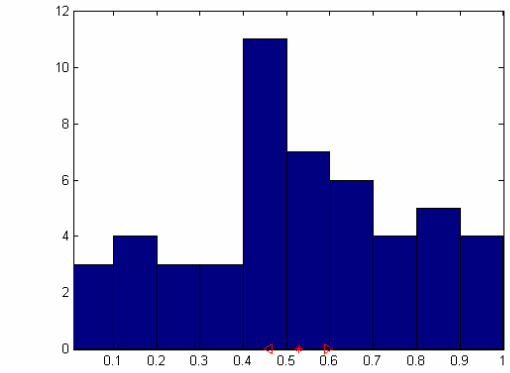
Fonema: /r/



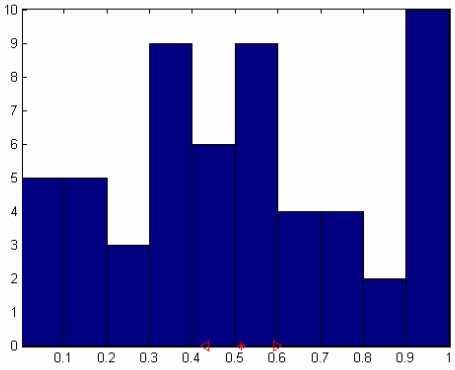
Fonema: /rr/



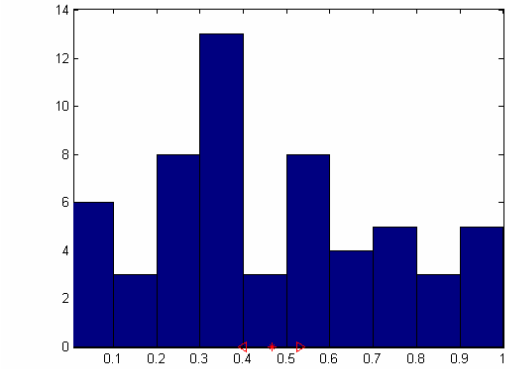
Fonema: /s/



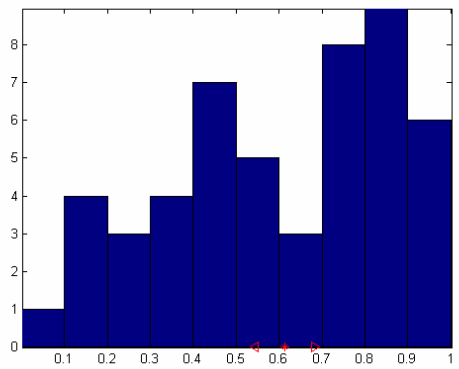
Fonema: /sil/



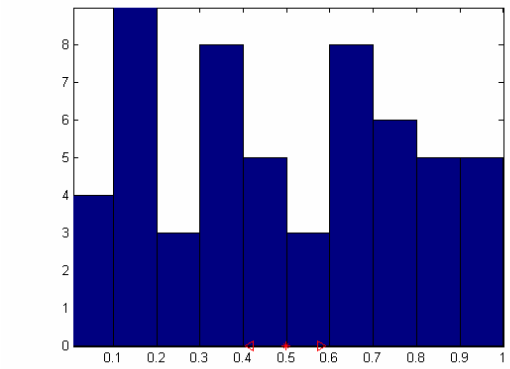
Fonema: /sp/



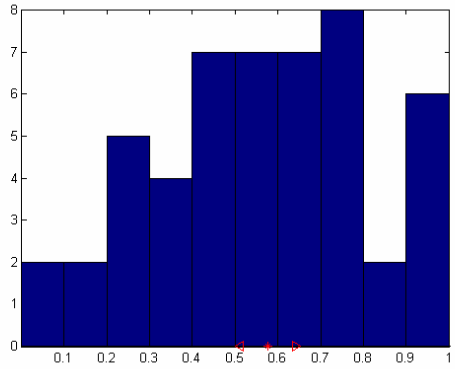
Fonema: /t/



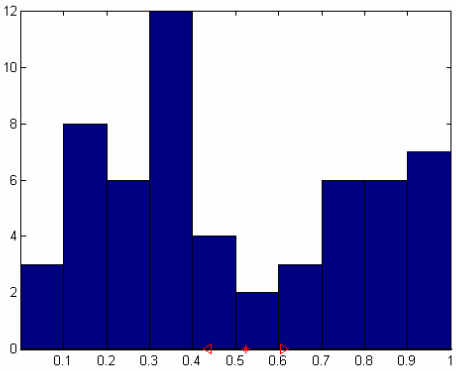
Fonema: /tS/



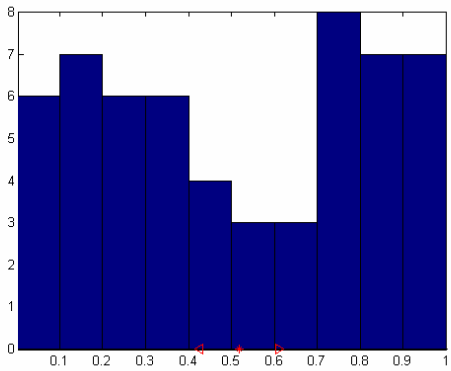
Fonema: /u/



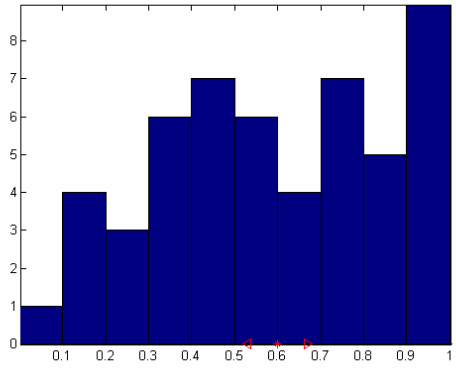
Fonema: /w/



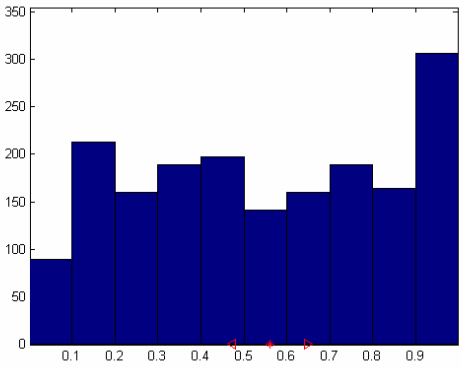
Fonema: /x/



Fonema: /z/



Fonema: TODOS



De las gráficas anteriores se puede deducir que no hay una regla clara a la hora de ubicar los SV aunque se aprecia una tendencia a colocarlos tanto en las posiciones centrales como en las finales de cada fonema.

En la siguiente tabla podemos ver como las medias, en la práctica totalidad de los casos, se encuentran por encima del 50% de la duración del fonema, lo que nos hace pensar que la máquina tiende a ubicar sus SVs en la segunda mitad del fonema. Esto concuerda con la hipótesis planteada al observar los histogramas de que la posición de los vectores soporte tiende a ser hacia la mitad o el final del fonema.

Fonema	/B/	/D/	/G/	/J/	/L/	/N/	/T/
Media	0,56022	0,6368	0,63521	0,48862	0,51699	0,62484	0,52153
Varianza	0,099966	0,10027	0,082512	0,077308	0,095727	0,083006	0,087686
Fonema	/a/	/b/	/d/	/e/	/f/	/g/	/i/
Media	0,52857	0,53807	0,61925	0,60268	0,52803	0,72809	0,61048
Varianza	0,082743	0,088011	0,087569	0,092003	0,079072	0,071469	0,09963
Fonema	/j/	/jj/	/k/	/l/	/m/	/n/	/o/
Media	0,6466	0,54244	0,56282	0,5307	0,50595	0,54075	0,56881
Varianza	0,090411	0,080431	0,074647	0,083675	0,074875	0,093871	0,12881
Fonema	/p/	/r/	/rr/	/s/	/sil/	/sp/	/t/
Media	0,59477	0,52645	0,51582	0,52695	0,51395	0,46533	0,61357
Varianza	0,08873	0,080104	0,090209	0,06712	0,083668	0,065097	0,06883
Fonema	/tS/	/u/	/w/	/x/	/z/		TOTAL
Media	0,49781	0,57556	0,52294	0,51794	0,59996		0,56086
Varianza	0,080973	0,064516	0,087389	0,091213	0,069285		0,087872

Tabla 3. Medias y varianzas en segmentado manual

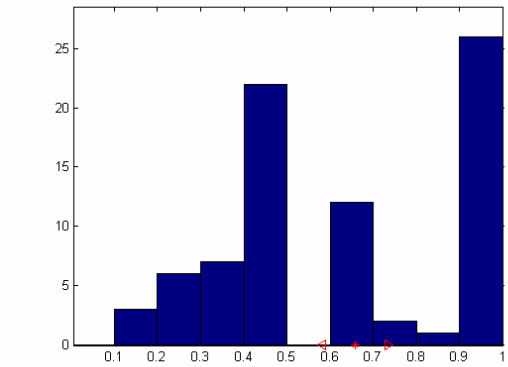
5.2. Comparación con segmentado híbrido HMM/SVM.

Como elemento de contraste con los resultados del apartado anterior, se eligieron las transiciones proporcionadas por el sistema híbrido HMM/SVM, para poder comparar los resultados de distintas técnicas.

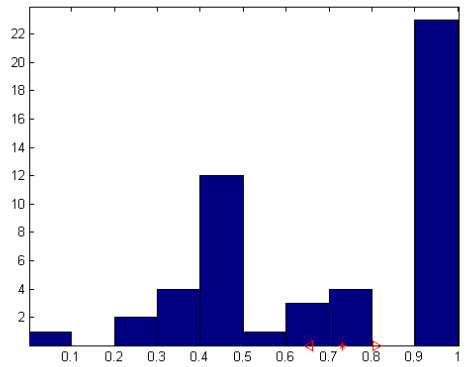
Cuando el sistema híbrido realiza la decodificación de una locución, etiqueta cada trama como perteneciente a una de las unidades acústicas consideradas. Por lo tanto, aprovechando esta característica se puede de una forma más o menos directa recuperar dichas etiquetas y determinar los instantes en los que se producen las transiciones.

Como se comentó en el apartado anterior, para poder realizar una comparación en las condiciones lo más similares posibles, los histogramas que se presentan a continuación están construidos en base a los mismos ficheros que se etiquetaron manualmente. No obstante, para completar el análisis, se realizó este mismo experimento con la totalidad de los ficheros, cuyos resultados se adjuntan en uno de los Anexos de este Proyecto.

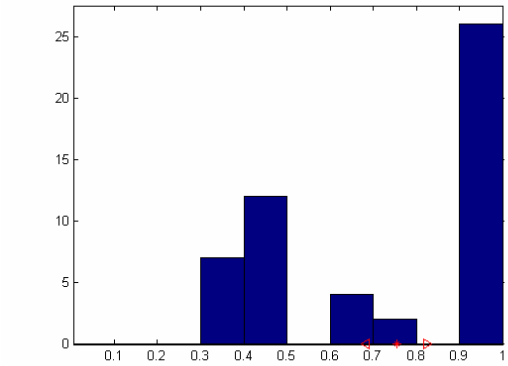
Fonema: /B/



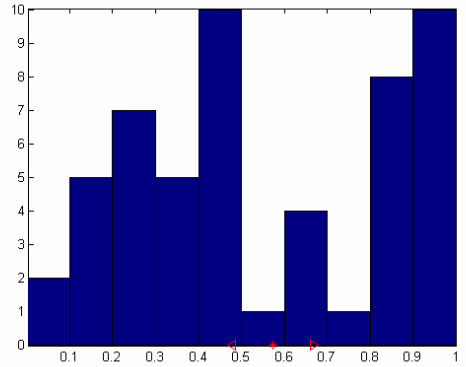
Fonema: /D/



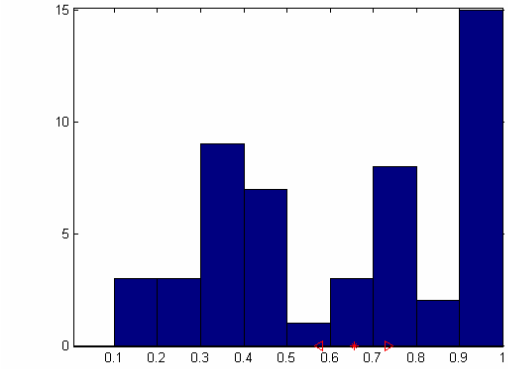
Fonema: /G/



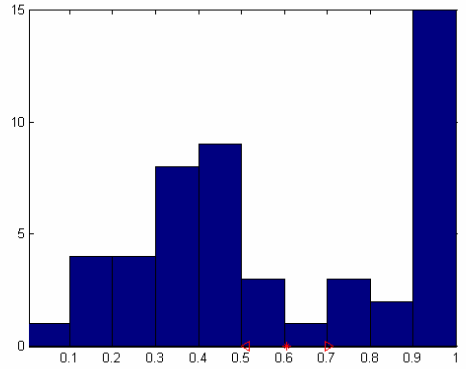
Fonema: /J/



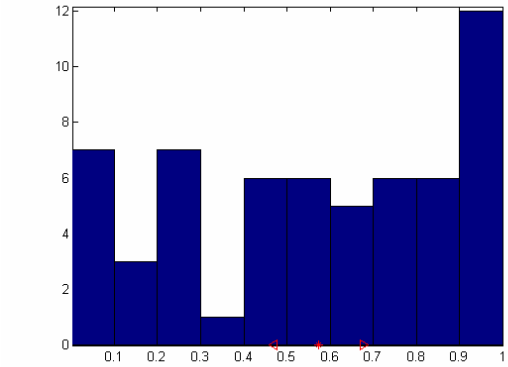
Fonema: /L/



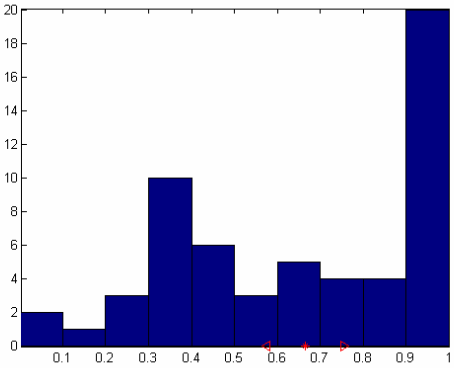
Fonema: /N/



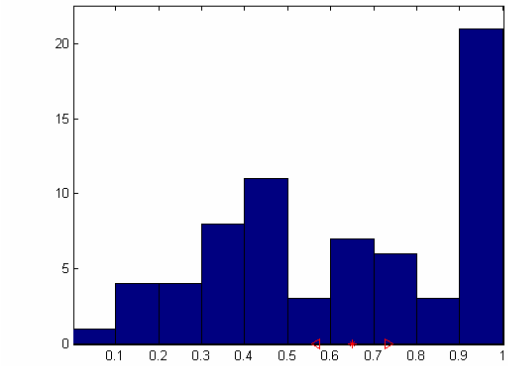
Fonema: /T/



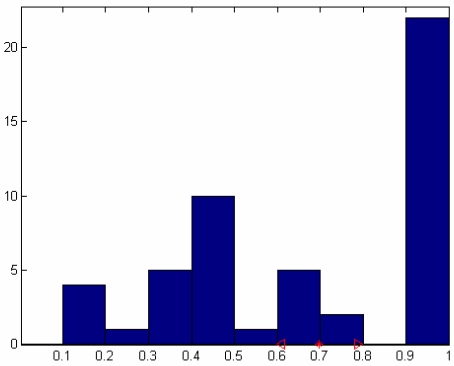
Fonema: /a/



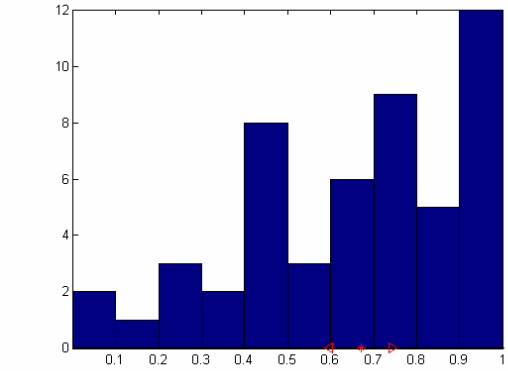
Fonema: /b/



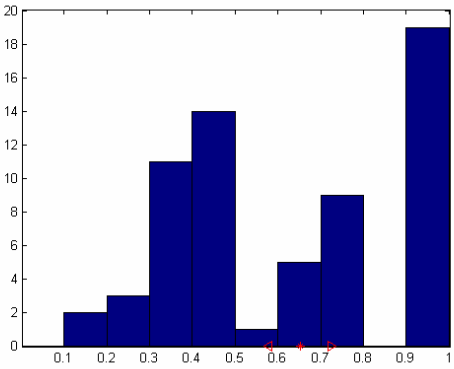
Fonema: /d/



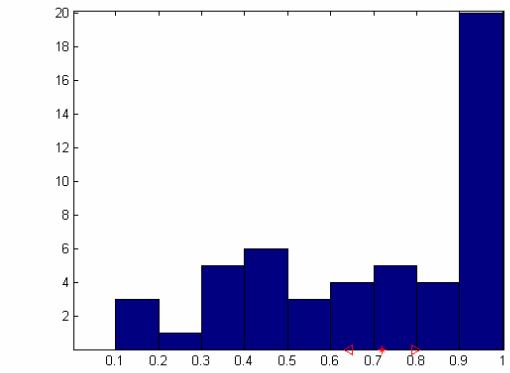
Fonema: /e/



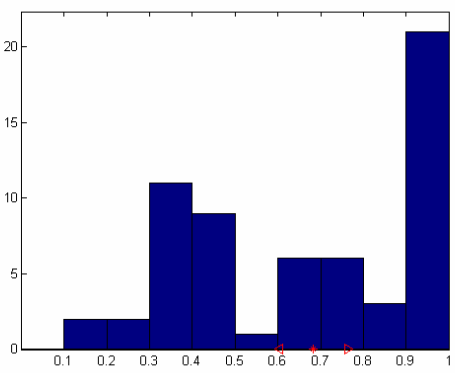
Fonema: /f/



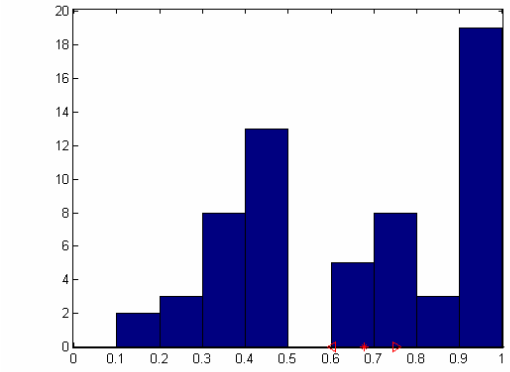
Fonema: /g/



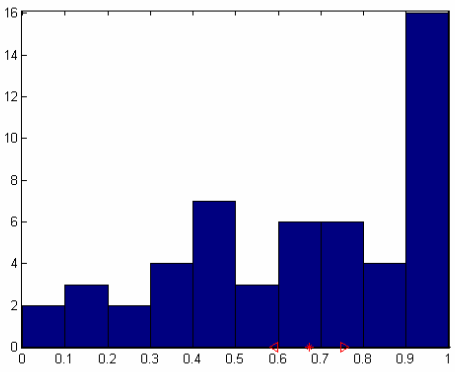
Fonema: /i/



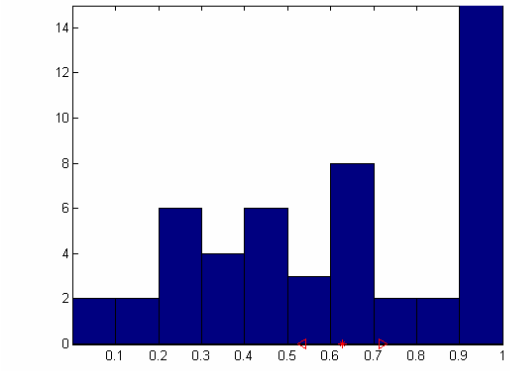
Fonema: /j/



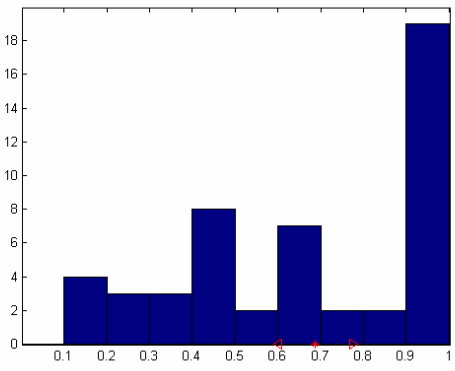
Fonema: /jj/



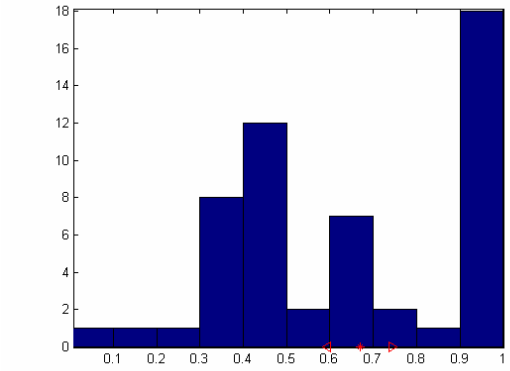
Fonema: /k/



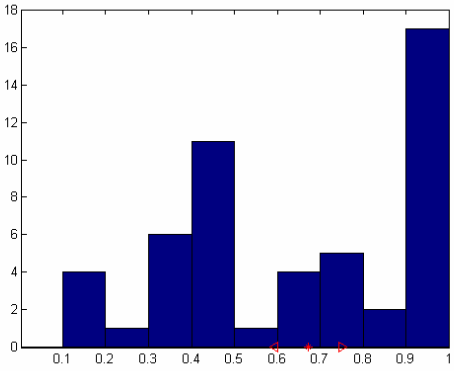
Fonema: /l/



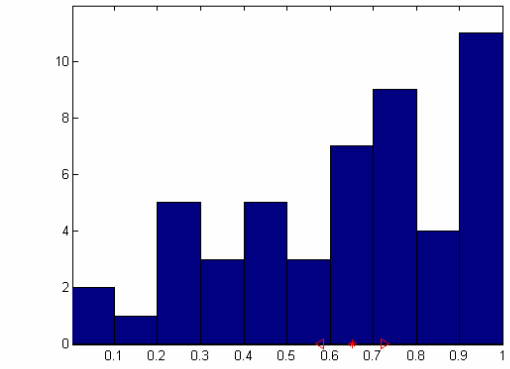
Fonema: /m/



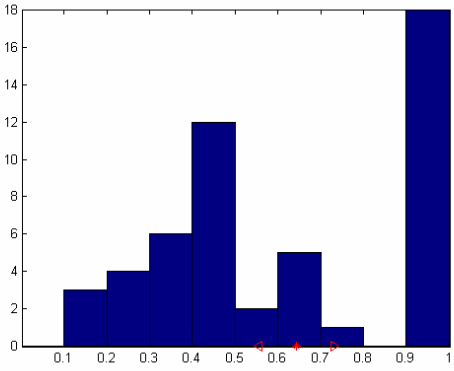
Fonema: /n/



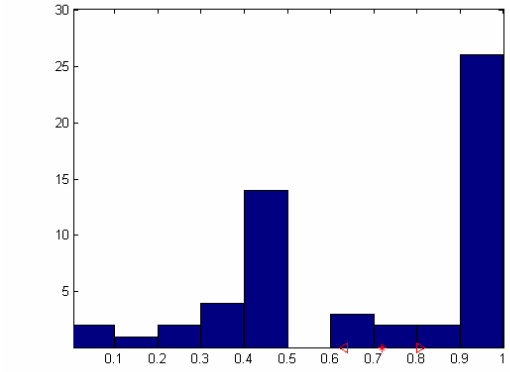
Fonema: /o/



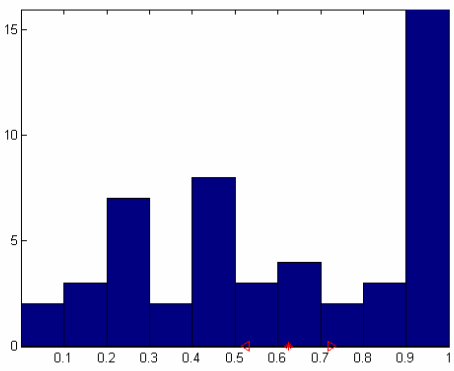
Fonema: /p/



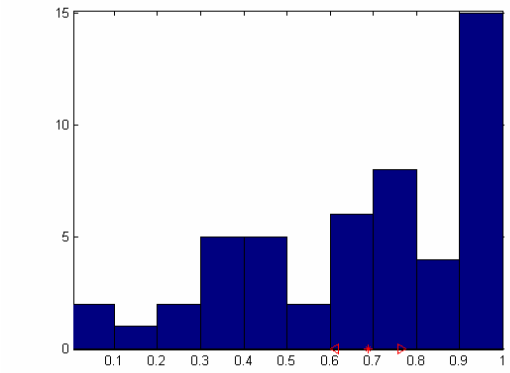
Fonema: /r/



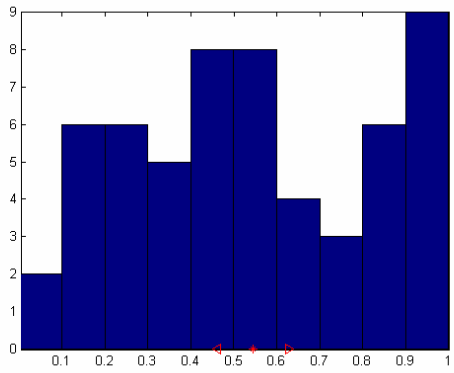
Fonema: /rr/



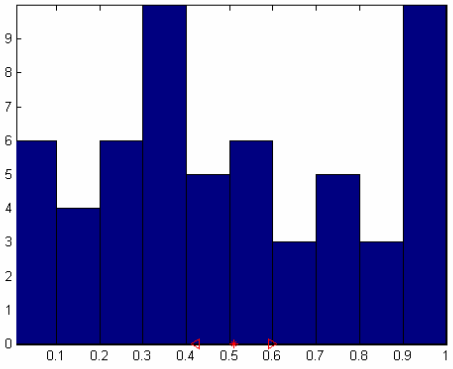
Fonema: /s/



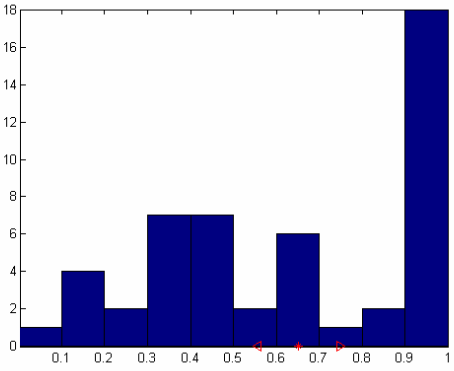
Fonema: /sil/



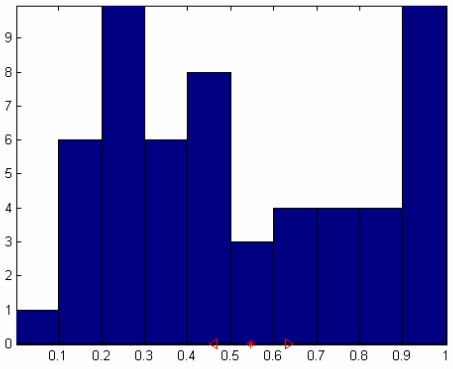
Fonema: /sp/



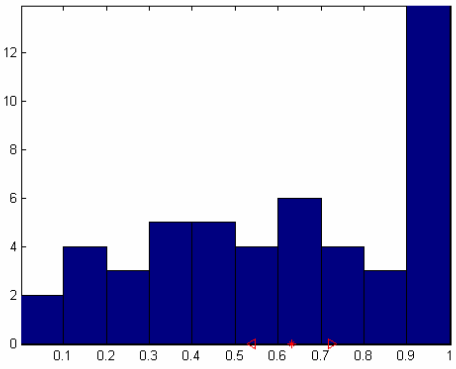
Fonema: /t/



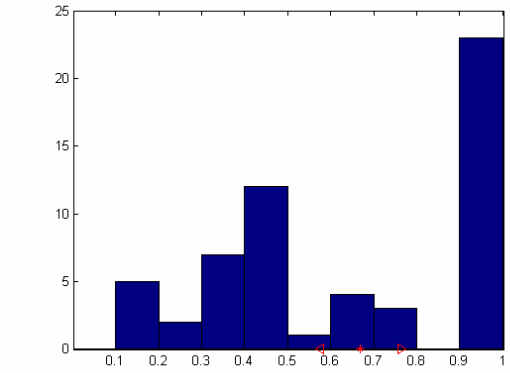
Fonema: /tS/



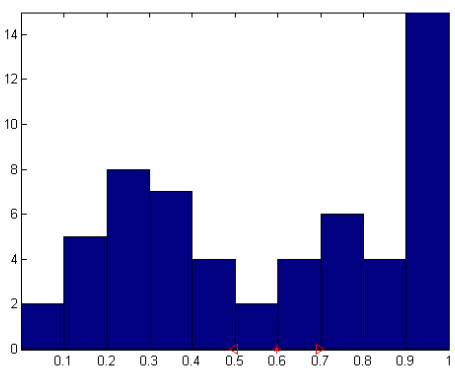
Fonema: /u/



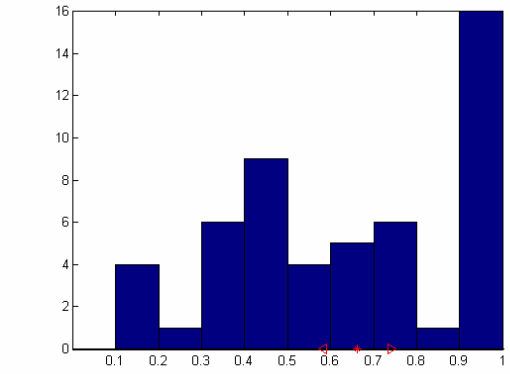
Fonema: /w/



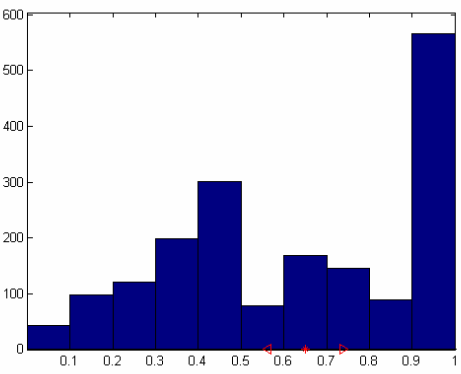
Fonema: /x/



Fonema: /z/



Fonema: TODOS



Con los datos vistos en los histogramas que se acaban de representar, se vuelve a apreciar que los vectores soporte se localizan principalmente al final del fonema. Pero en este caso se observa de una manera mucho más evidente. Además, dependiendo del fonema aparecen uno o dos picos secundarios centrados, fundamentalmente, en torno a la mitad de la duración del fonema.

Por último, examinando la tabla que se presenta a continuación, se ve como el mayor número de vectores soporte ubicados al final de los fonemas hace que la media en el valor de su posición aumente con respecto a los valores mostrados en el apartado anterior.

Fonema	/B/	/D/	/G/	/J/	/L/	/N/	/T/
Media	0,65761	0,73149	0,75383	0,57433	0,65486	0,6064	0,5736
Varianza	0,075659	0,077141	0,070673	0,094081	0,080074	0,09561	0,1032
Fonema	/a/	/b/	/d/	/e/	/f/	/g/	/i/
Media	0,66561	0,64987	0,69747	0,67037	0,65218	0,71974	0,68307
Varianza	0,089636	0,083616	0,088385	0,072003	0,072628	0,076589	0,080012
Fonema	/j/	/jj/	/k/	/l/	/m/	/n/	/o/
Media	0,67797	0,67314	0,62812	0,68818	0,66934	0,67216	0,65184
Varianza	0,07446	0,081354	0,092098	0,085535	0,074723	0,079081	0,073623
Fonema	/p/	/r/	/rr/	/s/	/sil/	/sp/	/t/
Media	0,64342	0,71865	0,62494	0,68734	0,54396	0,50939	0,65204
Varianza	0,087199	0,087407	0,10013	0,075844	0,082981	0,086923	0,095079
Fonema	/tS/	/u/	/w/	/x/	/z/		TOTAL
Media	0,54721	0,63253	0,66962	0,59733	0,66282		0,64971
Varianza	0,086537	0,091899	0,093326	0,099669	0,078339		0,087191

Tabla 4. Medias y varianzas en segmentado SVM

Examinando los histogramas se puede apreciar como la mayoría de ellos tiene una distribución de muestras muy similar, con un máximo principal muy marcado en torno al 1 y otro máximo secundario en torno al 0.5. Pero existen algunas anomalías que se tratarán de forma independiente:

- Fonema /J/: los dos máximos de los que se hablaba anteriormente contienen el mismo número de muestras, por lo que no cabría distinción entre máximo principal y secundario. La localización de éstos mantiene el patrón apreciado para el resto de fonemas entorno al 50% y 100% de la duración del fonema.
- Fonema /T/: sólo aparece un máximo principal al final del fonema, pero el resto de SVs se distribuye de forma más o menos uniforme en el resto del fonema.
- Fonema /e/: el máximo secundario aparece entre el 70% y 80% de la duración del fonema.
- Fonema /o/: al igual que con el fonema /e/, su máximo secundario se desplaza entre el 70% y el 80%.
- Fonema /sil/: se aprecia el máximo al final del fonema, pero la distribución de todos los SVs es bastante uniforme en todo el histograma.
- Fonema /sp/: al igual que con el fonema /J/, tenemos los dos máximos con el mismo número de muestras. Pero en este caso el primer máximo aparece al 30% de la duración del fonema. La distribución de SVs en el resto del fonema se mantiene más o menos uniforme.
- Fonema /tS/: el histograma de este fonema es muy similar al del /sp/.

Para contrastar estas anomalías debemos examinar los datos adjuntos en el Anexo A. En este anexo se muestran los mismos histogramas que en este apartado pero utilizando todas los ficheros de la base de datos, lo que nos proporcionará una información más general que la presentada en este punto.

A la vista de los histogramas del Anexo A vemos como las anteriores anomalías se reducen únicamente a tres fonemas: /sil/, /sp/ y /tS/, de cuyas características se hablará en el capítulo final de conclusiones.

Capítulo 6.
Conclusiones

Analizando los resultados del capítulo 5, se observa como en el análisis realizado empleando el segmentado obtenido del híbrido HMM/SVM existe una gran acumulación de SVs en la zona próxima al final del fonema y un pequeño máximo secundario en torno a la parte central del fonema. Esto se cumple para la práctica totalidad de las clases, pero existen tres excepciones que se tratan a continuación de forma individual:

- Fonema /sil/: Esta clase representa los silencios, los cuales no son propiamente un fonema y cuyas características, salvo en presencia de ruido, suelen ser muy homogéneas entre su inicio y su fin. Por lo tanto es lógico pensar que la distribución de los SVs en este fonema sea mucho más homogénea que para el resto de fonemas.

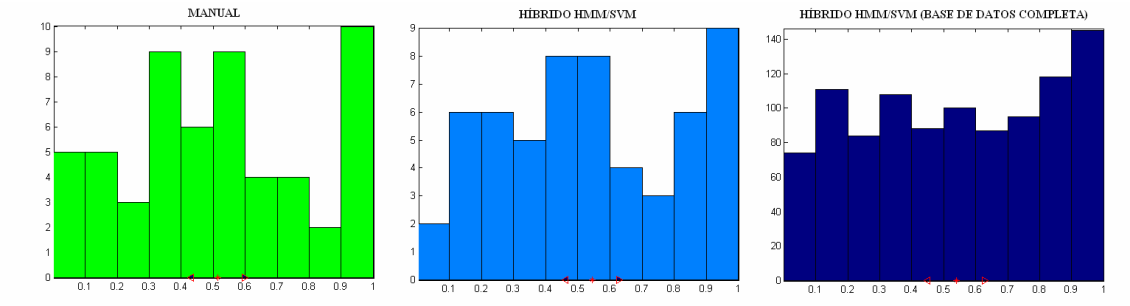


Figura 16. Comparación 3 segmentados /sil/

Extendiendo el análisis a los otros dos segmentados realizados, vemos que en los dos primeros da la sensación de no seguir una distribución homogénea, pero al introducir el segmentado del híbrido con la base de datos completa vemos como sí aparece esta característica.

- Fonema /sp/: Este fonema representa un silencio breve (‘short pause’), por lo que cabe esperar que su comportamiento, en cuanto a la ubicación de los SVs se refiere, sea lo más parecido posible al del silencio normal. Analizando el histograma obtenido en base a las transiciones del híbrido (apartado 5.2) se observan dos máximos, el primero en torno al 30-40% y el segundo en el 90-100% de la duración del fonema, pero para el resto de instantes la distribución es bastante similar.

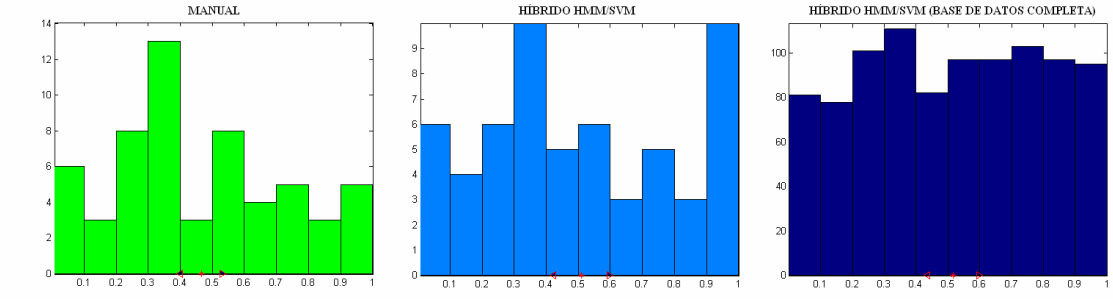


Figura 17. Comparación 3 segmentados /sp/

Cuando se introduce en el análisis el segmentado manual, sigue apareciendo el primer máximo lo que parece marcar alguna zona característica del fonema. Por otro lado, observando el segmentado del híbrido con la base de datos completa, se aprecia como aparece la homogeneidad que se espera de los silencios, pero también se sigue apreciando el primero de los máximos, por lo que parece que esa zona es representativa de este fonema en términos de la ubicación de SVs.

- Fonema /tS/: Este fonema de tipo alveolar africado suele transcribirse comúnmente por el grafema /ch/ y es un fonema de muy corta duración. Pero además la parte pura del fonema (libre de condicionamientos de los fonemas adyacentes) están en el primer tramo del mismo, donde se refleja un máximo, lo que podría indicar un posicionamiento en torno a una particularidad espectral propia de este fonema.

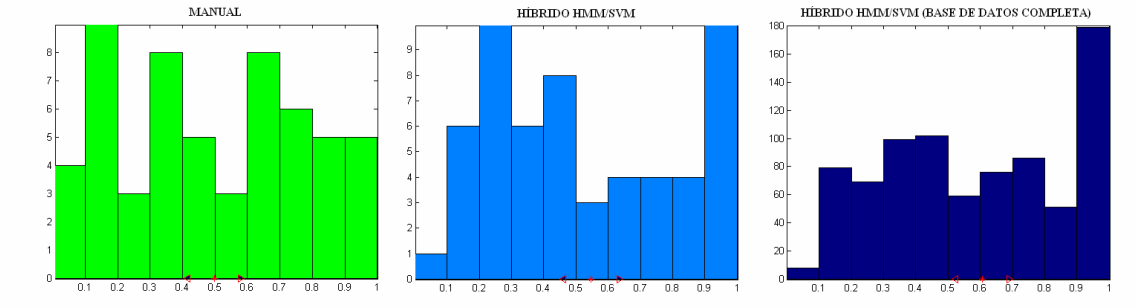


Figura 18. Comparación 3 segmentados /tS/

Al ampliar el análisis con el segmentado manual, parece corroborarse que el inicio del fonema es una zona representativa. Conclusión que no queda del todo clara cuando se observa el segmentado del híbrido con el total de la base de datos, donde únicamente aparece un máximo al final del fonema.

A la vista de los resultados obtenidos respecto al segmentado realizado por el híbrido HMM/SVM, parece que existe una tendencia en el entrenamiento del híbrido a seleccionar como SVs muestras que se sitúan en torno al 50% y 100% de la duración del fonema. Planteamiento que no queda del todo claro cuando se analizan los resultados obtenidos respecto al segmentado manual, donde en la mayoría de los fonemas parece mostrar una ausencia clara de patrones.

Por consiguiente procede, y queda como línea futura de trabajo, hacer un análisis de las diferencias observadas entre la segmentación manual y la automática a fin de determinar si las diferencias observadas obedecen a una falta de precisión del etiquetado de los sistemas empleados. Hasta que esto no se verifique, no es posible aseverar que los SVs se concentran en las zonas central y final de los fonemas, como parece deducirse del segmentado mediante el sistema híbrido.

Por todo lo anterior, como línea futura de investigación se propone el estudio con un tercer método de contraste, para el cual se propone elegir algún tipo de segmentado automático, sobre el cual ya se realizó algún intento durante la elaboración de este proyecto, pero el cual no dio resultados correctos en la determinación de las transiciones entre fonemas. Una vez obtenido este método automático, y si quedara contrastada la tendencia que se ha expuesto en los párrafos anteriores, sería de interés estudiar si las

prestaciones de este tipo de híbridos mejoraría forzando la selección de SVs en estas zonas representativas de los fonemas.

En lo que respecta a la herramienta gráfica, cabe destacar el buen funcionamiento de la misma, ya que cumple con los objetivos para los que fue diseñada. El estudio visual de los SVs se realiza de una manera fácil e intuitiva, permitiéndonos en los casos en que fuera necesario ampliar la zona que se desea examinar, bien por haber una acumulación de SVs, bien por tratarse de una zona donde no está clara la transición entre los fonemas adyacentes.

Pese a que la herramienta cumple perfectamente con su cometido, se debe mencionar la posibilidad de desarrollo sobre la misma, cuyo manual de uso y código se aporta en los Anexos B y C de este Proyecto Fin de Carrera. Algunos puntos de desarrollo que se proponen para ella son:

- Integración con las máquinas de vectores soporte: una característica que añadiría valor a esta herramienta sería su integración con el proceso de entrenamiento y test de la SVM, permitiendo dirigir y controlar estos procesos desde la propia interfaz.

- Nuevas características en la reproducción del audio: otra posible mejora sobre la herramienta sería mostrar sobre el área donde se representan las figuras una barra de progreso que indicara qué instante de la señal se está reproduciendo.

- Integración de un detector automático de transiciones, dado que anteriormente se comentó la necesidad de un tercer método de contraste y además se proponía que éste realizara el segmentado de forma automática. Sería apropiado que, en caso de que este nuevo módulo diera unos resultados satisfactorios, pudiera ser integrado en esta herramienta gráfica.

Capítulo 7.
Presupuesto

En este último capítulo del Proyecto Fin de Carrera se realizará un presupuesto del mismo. Para ello, en primer lugar se hará una descripción de las tareas que lo componen, para después finalizar con la valoración económica.

7.1. Fases del Proyecto.

FASE 1: Definición del proyecto.

- *Descripción:* En esta primera fase se plantearon los objetivos que se deseaban obtener con este PFC, así como las líneas de estudio que se podrían seguir para alcanzar tales objetivos.
- *Duración:* 14 horas.
- *Tareas:* Esta fase se puede descomponer en dos tareas que se denotarán como 1.a y 1.b, que se muestran a continuación.

Tarea 1.a: Definición de los objetivos.

- *Objetivos:* En este punto se definieron los objetivos principales que se pretendían conseguir con este PFC, como son la implementación de una herramienta visual, así como la obtención de unos resultados estadísticos para el análisis de la localización de los SVs.
- *Duración:* 8 horas.

Tarea 1.b: Definición de la organización/programación.

- *Objetivos:* Establecer los pasos que se deben seguir en el desarrollo de este PFC y la dependencia entre las distintas tareas que los componen, para obtener los fines pretendidos.
- *Duración:* 6 horas.

FASE 2: Documentación.

- *Descripción:* Adquirir la base teórica necesaria sobre las tecnologías empleadas, así como los conocimientos sobre las herramientas que se emplearán en la elaboración del PFC.
- *Duración:* 230 horas.
- *Tareas:* Se descompone la fase 2 en dos tareas principales: 2.a y 2.b.

Tarea 2.a: Documentación teórica.

- *Objetivos:* Se pretende adquirir los fundamentos necesarios sobre el funcionamiento de las tecnologías que se aplican actualmente en el marco del reconocimiento automático del habla, para afrontar los estudios que se llevarán a cabo en este PFC. Estas tecnologías son los HMMs, ANNs y SVMs.
- *Duración:* 70 horas.

Tarea 2.b: Estudio de las herramientas a emplear.

- *Objetivos:* En el desarrollo de este PFC se emplean distintas herramientas, sobre las cuales es necesario tener cierto conocimiento sobre su funcionamiento. Estas herramientas son HTK, libSVM, Matlab, Cool Edit.
- *Duración:* 160 horas.

FASE 3: Programación de software.

- *Descripción:* En esta fase del PFC se crean las nuevas funciones y se adaptan los scripts que se utilizarán en el resto de fases.
- *Duración:* 200 horas.
- *Tareas:* Son tres las tareas en las que se puede dividir esta fase: 3.a, 3.b y 3.c.

Tarea 3.a: Programación para el RAH.

- *Objetivos:* Modificación y compilación de los scripts que realizarán tanto el entrenamiento como el test del híbrido HMM/SVM y las variables que lo componen para su correcta ejecución.
- *Duración:* 20 horas.

Tarea 3.b: Programación de funciones auxiliares.

- *Objetivos:* Definición de nuevas funciones en Matlab que nos permitan modificar los ficheros de audio a un formato que Matlab pueda reproducir, funciones que transformen los ficheros de audio en ficheros '.mat' con los que se pueda trabajar fácilmente en Matlab, funciones que sean capaces de extraer todos los SVs de los ficheros '.model' que nos devolverá el híbrido HMM/SVM, funciones que sean capaces de buscar los SVs dentro de la base de datos, funciones que nos procesen los datos obtenidos y nos faciliten los datos para realizar el estudio estadístico.
- *Duración:* 30 horas.

Tarea 3.c: Programación de la herramienta gráfica.

- *Objetivos:* Definir las funciones necesarias para el funcionamiento de la herramienta visual, así como la configuración de los distintos elementos dentro de la misma para ofrecer al usuario un entorno lo más sencillo posible.
- *Duración:* 150 horas.

FASE 4: Experimentos y obtención de resultados.

- *Descripción:* En esta fase abarca los procesos de entrenamiento y test del híbrido HMM/SVM, así como la obtención de los distintos segmentados realizados sobre la base de datos, en definitiva todos las tareas necesarias para la recopilación de los datos necesarios para obtener los resultados del PFC.
- *Duración:* 300 horas.
- *Tareas:* Esta fase se descompone en 6 tareas: 4.a, 4.b, 4.c, 4.d, 4.e, y 4.f.

Tarea 4.a: Entrenamiento del híbrido HMM/SVM.

- *Objetivos:* Buscar los parámetros óptimos de 'C' y 'G' mediante validación cruzada y realizar el entrenamiento de la SVM con los parámetros óptimos conseguidos.
- *Duración:* 60 horas.

Tarea 4.b: Reconocimiento mediante híbrido HMM/SVM.

- *Objetivos:* Con los parámetros óptimos hallados en la tarea anterior, realizar el reconocimiento sobre la base de datos de test.
- *Duración:* 20 horas.

Tarea 4.c: Búsqueda de SVs.

- *Objetivos:* Partiendo del fichero '.model' que nos devuelve el entrenamiento del híbrido, se deben recuperar todos los SVs y poder determinar a que fichero de la base de datos corresponde y su posición dentro de él.
- *Duración:* 20 horas.

Tarea 4.d: Segmentado manual de la base de datos.

- *Objetivos:* En primer lugar se hace una selección sobre los ficheros de la base de datos, tratando que el número de SVs pertenecientes a cada fonema sea aproximadamente el mismo, pero estableciendo un umbral mínimo de SVs por fonema. Y en segundo lugar, empleando la herramienta Cool Edit, realizar una localización manual de las transiciones correspondientes a los ficheros seleccionados de la base de datos.
- *Duración:* 120 horas.

Tarea 4.e: Segmentado de la base de datos realizado por el híbrido HMM/SVM.

- *Objetivos:* En esta tarea se deberán procesar los ficheros de reconocimiento proporcionados por el híbrido HMM/SVM para obtener las transiciones entre fonemas que halla considerado en el proceso de reconocimiento.
- *Duración:* 20 horas.

Tarea 4.f: Obtención de resultados.

- *Objetivos:* Recopilar los datos obtenidos en el resto de tareas de la fase 4 y procesarlos de forma apropiada para obtener los resultados sobre los que realizar las valoraciones y conclusiones necesarias.
- *Duración:* 60 horas.

FASE 5: Redacción de la memoria.

- *Descripción y Objetivos:* Esta fase trata de recopilar toda la información obtenida durante la realización de este PFC, tanto datos teóricos como resultados experimentales, y presentarla en el formato adecuado para la exposición y defensa del trabajo realizado.
- *Duración:* 220 horas.
- *Tareas:* Esta fase no se descompone en más tareas.

7.2. Valoración Económica.

A continuación se muestra el desglose de los gastos derivados de la realización del presente proyecto fin de carrera:

- *Gastos de Personal:*

Para la realización de este proyecto se ha requerido a un Ingeniero Técnico en Telecomunicación (Proyectando) y un Ingeniero Superior en Telecomunicación (Tutor).

La tabla que se presenta a continuación contiene el resumen de las tareas realizadas y su duración:

CONCEPTO	DURACIÓN
FASE 1. Definición del Proyecto	14 horas
1.a. Definición de los Objetivos	*8 horas*
1.b. Definición de la Organización/Programación	*6 horas*
FASE 2. Documentación	230 horas
2.a. Documentación teórica	*70 horas*
2.c. Estudio de las tecnologías a emplear	*160 horas*
FASE 3. Programación de Software	200 horas
3.a. Programación para el RAH	*20 horas*
3.b. Programación de funciones auxiliares	*30 horas*
3.c. Programación de herramienta gráfica	*150 horas*
FASE 4. Experimentos y Obtención de Resultados	300 horas
4.a. Entrenamiento del híbrido HMM/SVM	*60 horas*
4.b. Reconocimiento mediante híbrido HMM/SVM	*20 horas*
4.c. Búsqueda de Vectores Soporte	*20 horas*
4.d. Segmentado manual	*120 horas*
4.e. Segmentado híbrido HMM/SVM	*20 horas*
4. f. Obtención de resultados	*60 horas*
FASE 5. Redacción de la Memoria	250 horas
TOTAL	*994 horas*

Tabla 5. Distribución en horas del Proyecto

De los datos mostrados en la tabla 5, el 18% del tiempo fue compartido junto al tutor del proyecto, con lo que los gastos finales de personal son:

GASTOS DE PERSONAL			
CONCEPTO	**HORAS**	**PRECIO (€/HORA)**	**COSTE**
Ingeniero Técnico	994	40,00	39.760,00
Ingeniero Superior	178,92	60,00	10.735,20
TOTAL			*50.495,20*

Tabla 6. Gastos de Personal

- *Gastos de Material:*

En este apartado se incluye gastos tales como equipo informático empleado, adquisición de software y similares.

GASTOS DE MATERIAL	
CONCEPTO	**COSTE**
Ordenador Personal	750,00 €
Licencia Matlab (uso comercial)	1.950,00 €
Documentación	160,00 €
Material de Oficina	120,00 €
TOTAL	*2.980,00 €*

Tabla 7. Gastos de Material

- *Presupuesto Total:*

En la siguiente tabla se adjunta el coste total del proyecto:

PRESUPUESTO TOTAL	
CONCEPTO	**COSTE**
Gastos de Personal	50.495,20 €
Gastos de Material	2.980,00 €
Base Imponible	*53.475,20 €*
I.V.A. (16%)	8.556,03 €
TOTAL	***62.031,23 €***

Tabla 8. Presupuesto Total del Proyecto

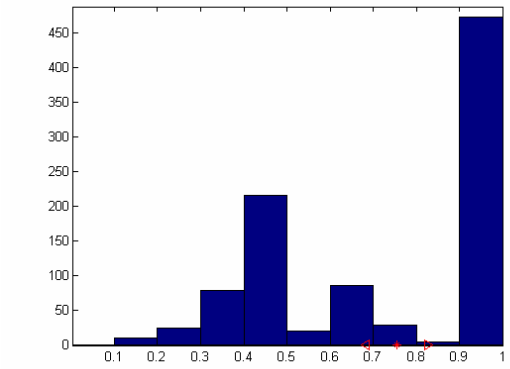
Por lo tanto, el coste total de este proyecto asciende a SESENTA Y DOS MIL TREINTA Y UN EUROS CON VEINTITRÉS CÉNTIMOS DE EURO.

Anexos

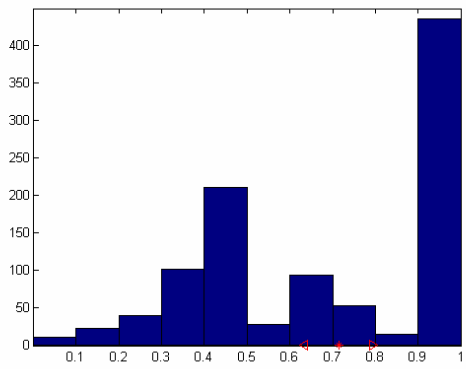
A. Comparación con segmentado SVM (sobre la base de datos completa).

En este anexo del proyecto se presentan los resultados obtenidos en la comparación con el segmentado de voz realizado por el híbrido HMM/SVM sobre el total de los ficheros que componen la base de datos. En primer lugar se mostrarán los histogramas obtenidos y a continuación se presenta una tabla resumen con los valores de media y varianza en la ubicación de los SV, tal y como se ha hecho con los apartados del capítulo 5.

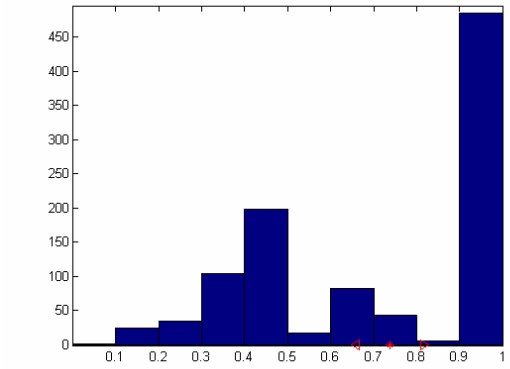
Fonema: /B/



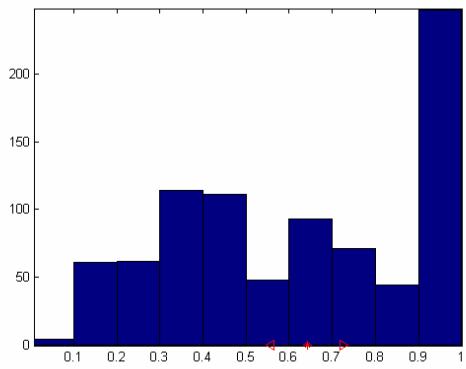
Fonema: /D/



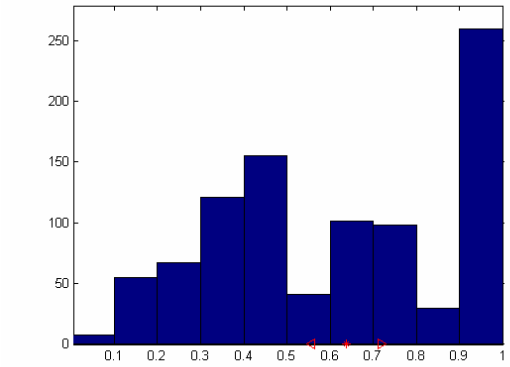
Fonema: /G/



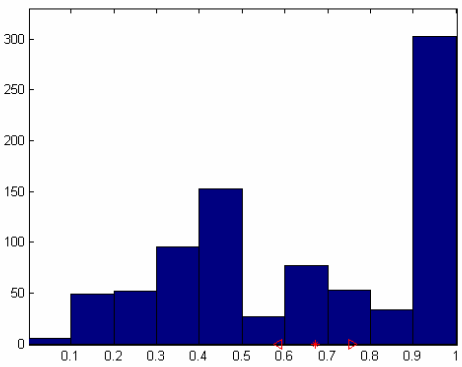
Fonema: /J/



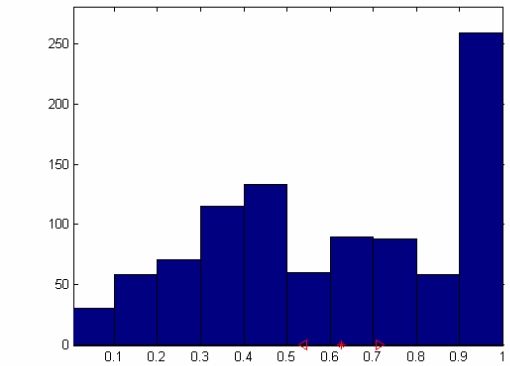
Fonema: /L/



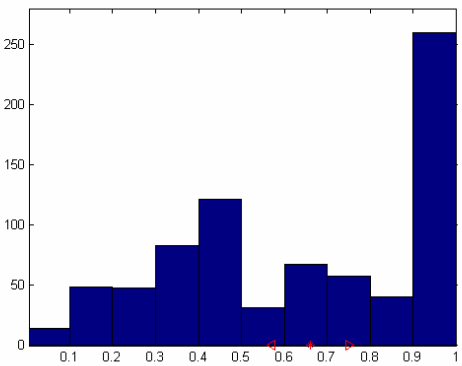
Fonema: /N/



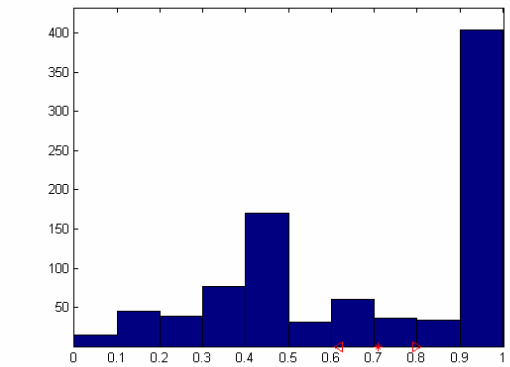
Fonema: /T/



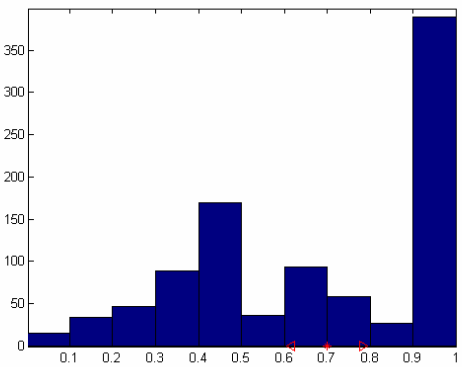
Fonema: /a/



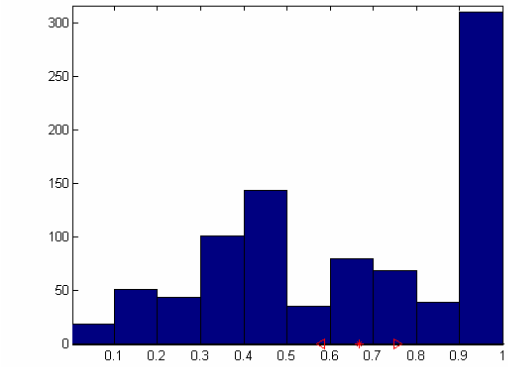
Fonema: /b/



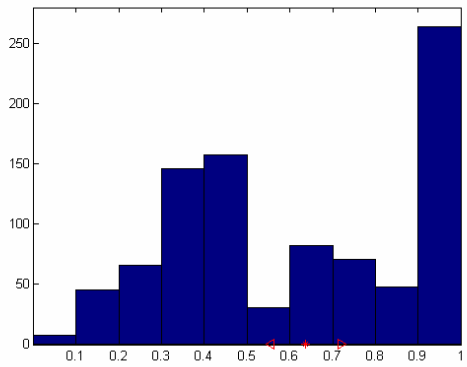
Fonema: /d/



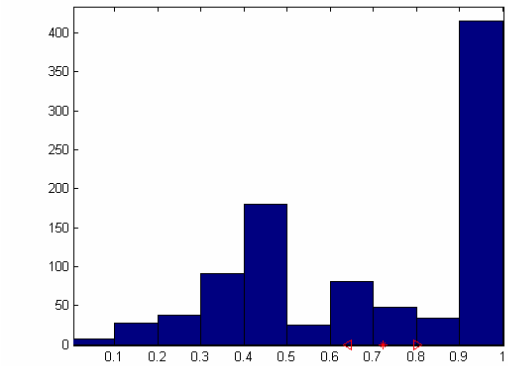
Fonema: /e/



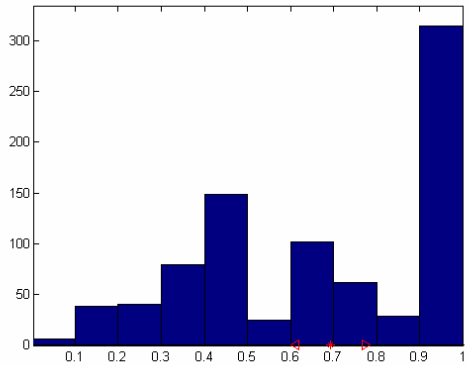
Fonema: /f/



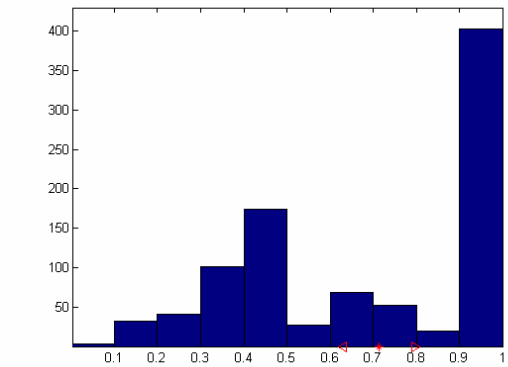
Fonema: /g/



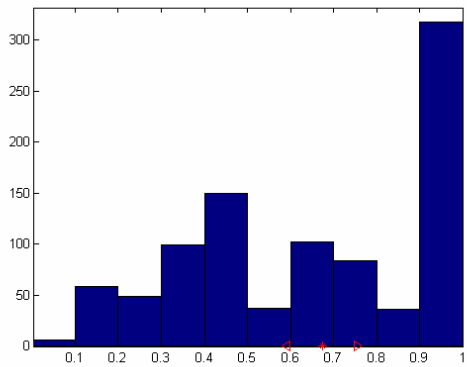
Fonema: /i/



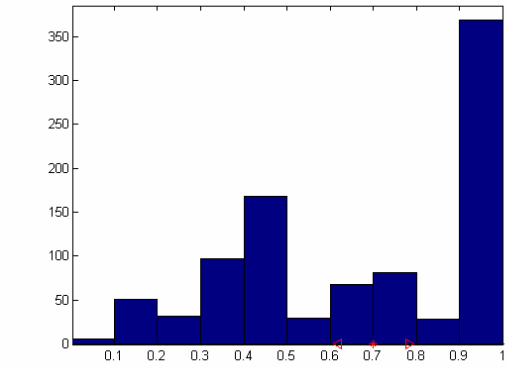
Fonema: /j/



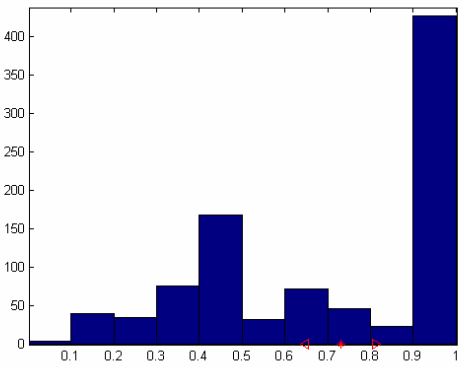
Fonema: /jj/



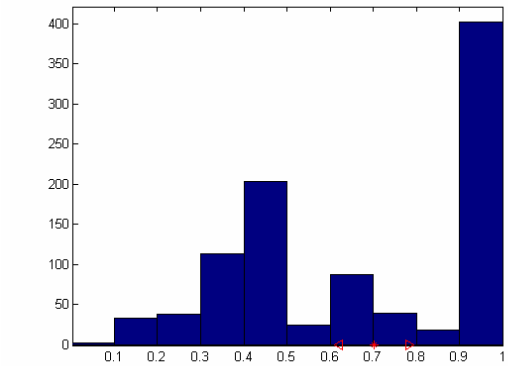
Fonema: /k/



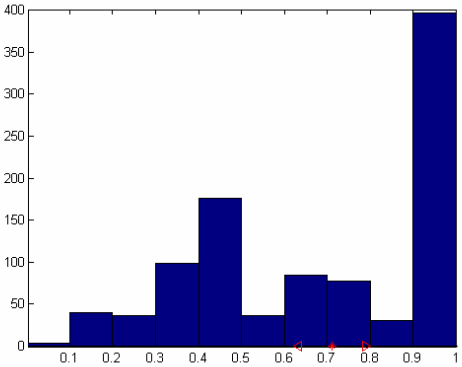
Fonema: /l/



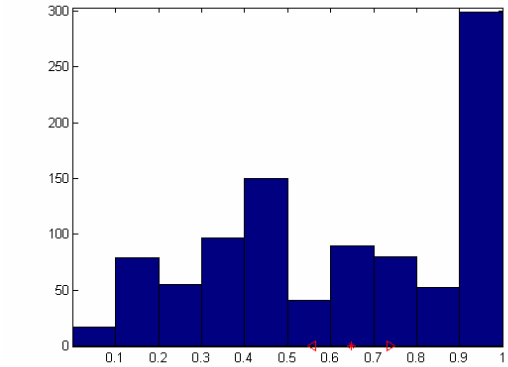
Fonema: /m/



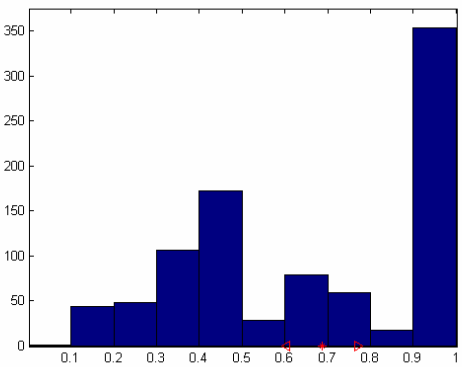
Fonema: /n/



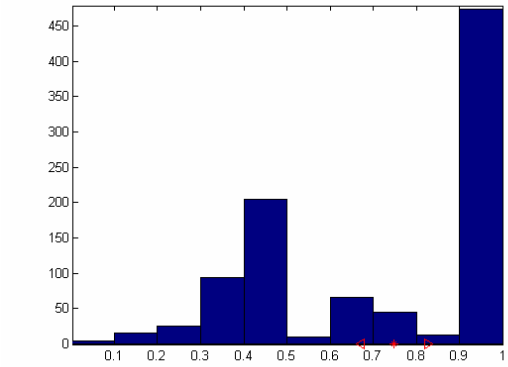
Fonema: /o/



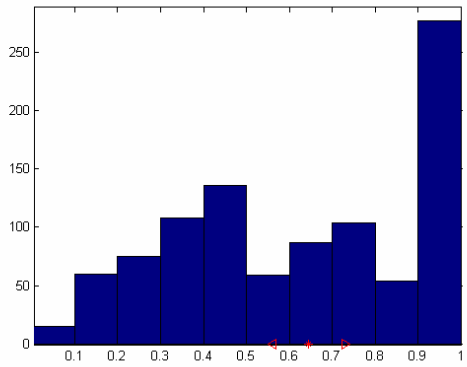
Fonema: /p/



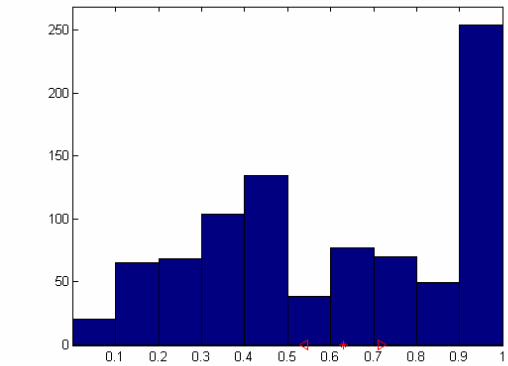
Fonema: /r/



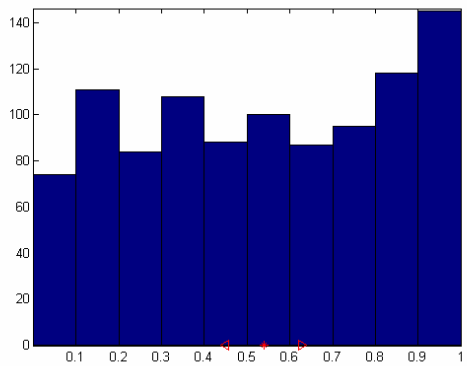
Fonema: /rr/



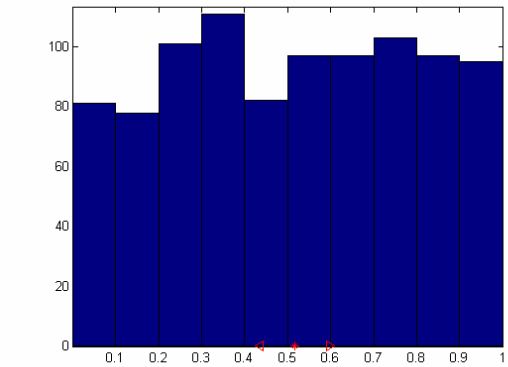
Fonema: /s/



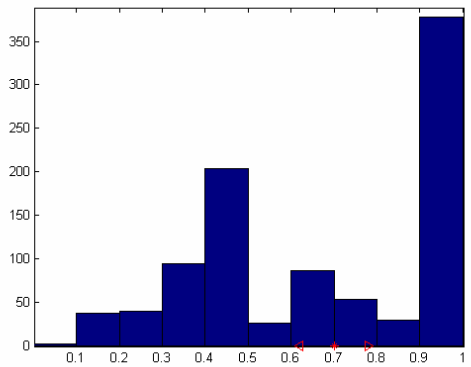
Fonema: /sil/



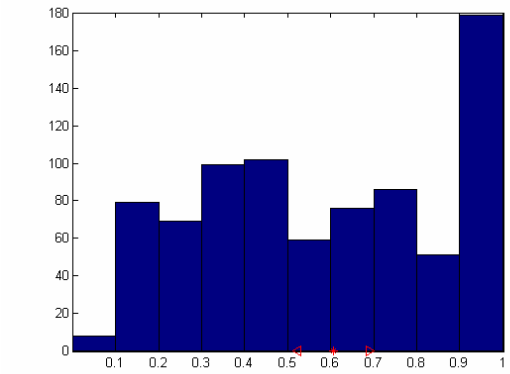
Fonema: /sp/



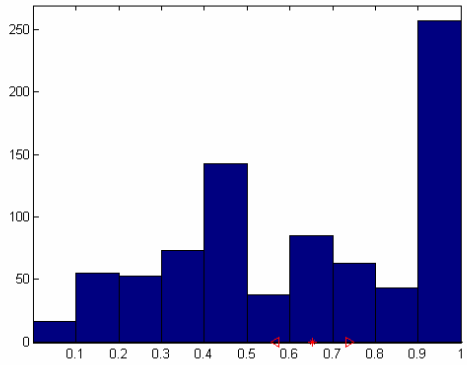
Fonema: /t/



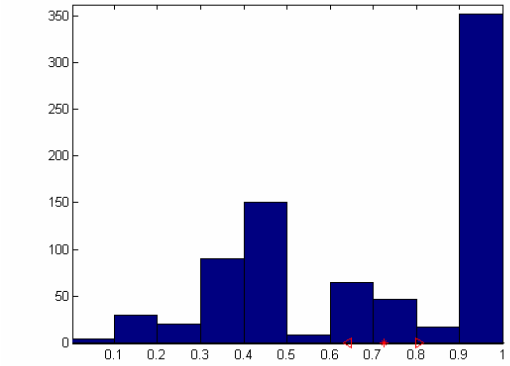
Fonema: /tS/



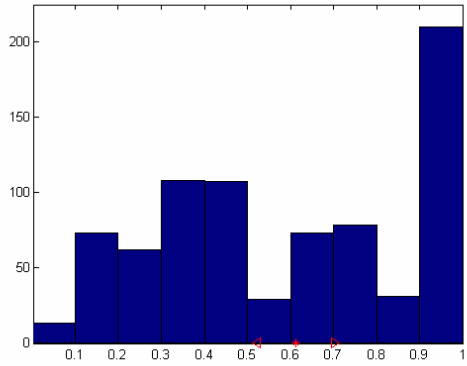
Fonema: /u/



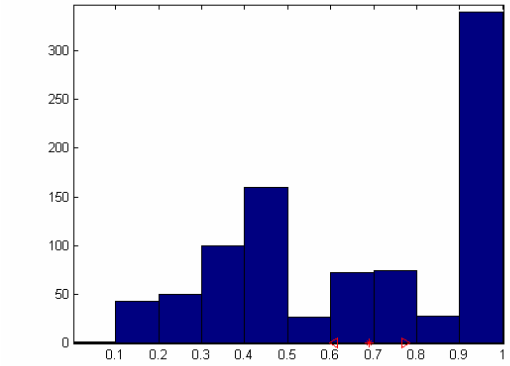
Fonema: /w/



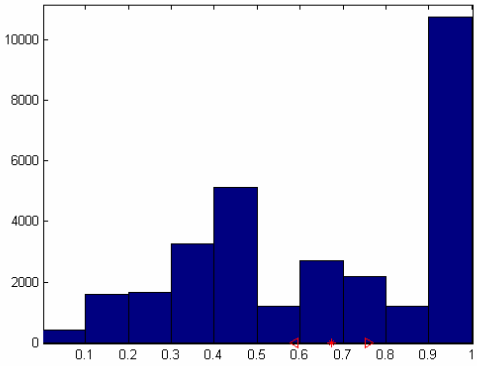
Fonema: /x/



Fonema: /z/



Fonema: TODOS



Fonema	/B/	/D/	/G/	/J/	/L/	/N/	/T/
Media	0,75505	0,71416	0,73764	0,64191	0,63804	0,67062	0,62563
Varianza	0,071106	0,079146	0,078663	0,084014	0,08002	0,085699	0,087325
Fonema	/a/	/b/	/d/	/e/	/f/	/g/	/i/
Media	0,6609	0,70819	0,69972	0,66838	0,6372	0,72208	0,69253
Varianza	0,088953	0,089207	0,083085	0,087318	0,082577	0,079251	0,080305
Fonema	/j/	/jj/	/k/	/l/	/m/	/n/	/o/
Media	0,714	0,67417	0,70063	0,7302	0,70233	0,71127	0,64735
Varianza	0,082191	0,081444	0,082069	0,081612	0,080763	0,078763	0,089344
Fonema	/p/	/r/	/rr/	/s/	/sil/	/sp/	/t/
Media	0,6872	0,74889	0,64524	0,62899	0,53897	0,51654	0,70062
Varianza	0,082935	0,076355	0,083552	0,089115	0,089068	0,08084	0,07941
Fonema	/tS/	/u/	/w/	/x/	/z/		TOTAL
Media	0,60658	0,65251	0,72437	0,61213	0,69053		0,67287
Varianza	0,082687	0,085953	0,081199	0,089368	0,08205		0,085761

Tabla 9. Medias y varianzas en segmentado SVM sobre el total de la BBDD

A la vista de los resultados de este anexo todo parece indicar que los vectores soporte escogidos por la SVM para representar los modelos de cada clase tienden a concentrarse en torno a la mitad y final del fonema.

B. Manual de uso de la herramienta gráfica.

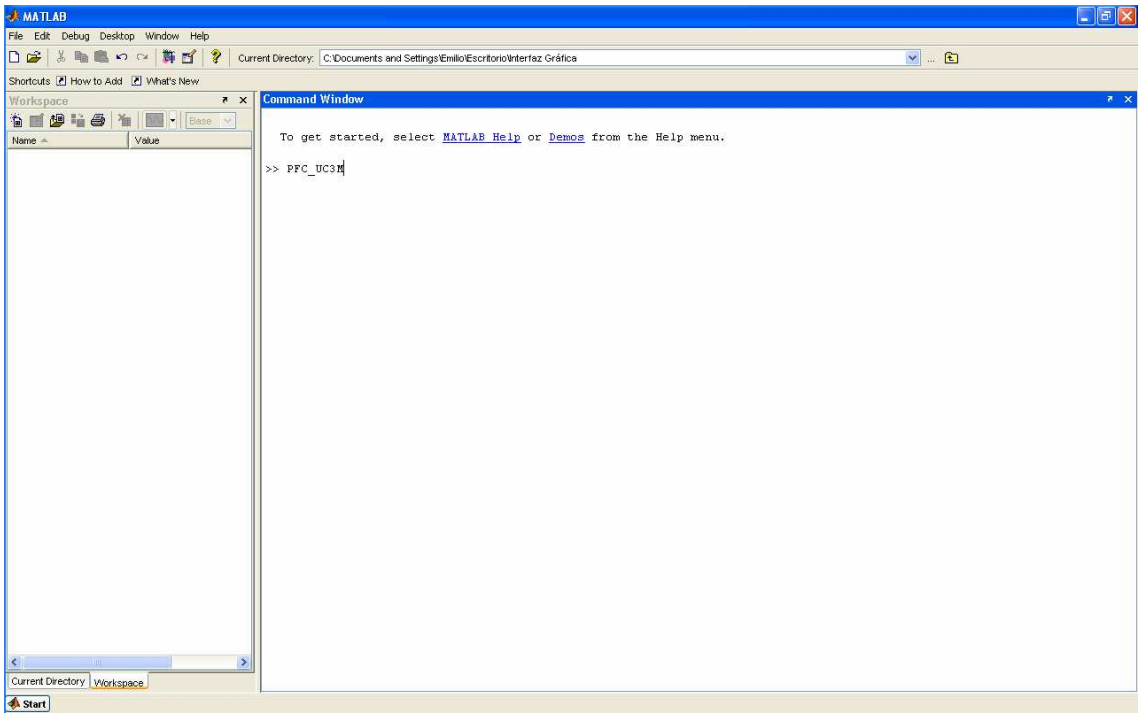
Este anexo pretende describir de forma detallada los pasos que se han de seguir para el uso de la herramienta gráfica.

Requisitos del sistema:

Dado que la GUI está implementada sobre Matlab, el requisito mínimo que debe cumplir el PC donde se quiera ejecutar la aplicación debe ser que tenga instalada esta aplicación. Se recomienda la versión 7.0 de este programa, ya que es la versión sobre la que se ha programado la herramienta. Aunque dada la filosofía de Matlab, en la que nuevas versiones del programa reutilizan las funciones ya implementadas en versiones previas, esta interfaz gráfica debería funcionar sin problemas en cualquier versión posterior.

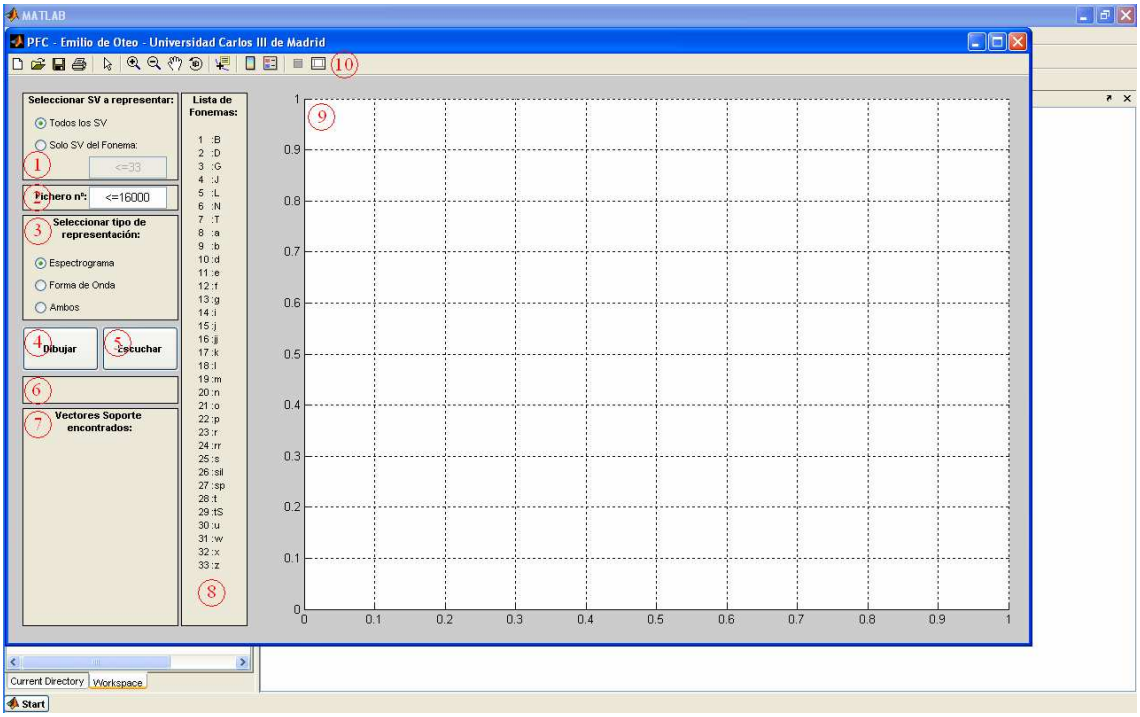
Ejecución de la Herramienta:

Una vez iniciado el Matlab y situados en el directorio donde tenemos la herramienta gráfica, sólo hay que situarse en la línea de comandos y ejecutar 'PFC_UC3M', tal y como se muestra en la siguiente captura:



Elementos de la Herramienta Gráfica:

La ejecución de este comando dará como resultado la aparición de la ventana correspondiente a la GUI.



Los puntos marcados en rojo de la imagen anterior se corresponden con los elementos principales de la herramienta los cuales se comentan a continuación:

1. *Seleccionar SV a representar*: este panel nos permite elegir si queremos representar todos los SVs que aparezcan en este fichero o sólo los de una determinada clase. Por defecto aparece marcada la primera opción, pero pulsando sobre la segunda se habilita el cuadro de texto que nos permite seleccionar la clase del fonema que buscamos (ver punto 8).

2. *Selección de fichero*: este elemento es un cuadro de texto donde se debe marcar el fichero que se desea representar; se debe rellenar de forma obligatoria, ya que sin él la aplicación devolverá un mensaje de error.

3. *Seleccionar tipo de representación*: panel que nos muestra las tres opciones de representación gráfica que permite la GUI. Estas tres opciones son ‘Espectrograma’ (opción por defecto), ‘Forma de Onda’ y ‘Ambas’. Esta última mostrará por pantalla las dos primeras de forma simultánea.

4. *Dibujar*: este botón ejecuta la orden de presentar por pantalla, haciendo que la herramienta recopile toda la información seleccionada en los distintos paneles y la muestre en el área destinada a tal efecto (ver punto 9).

5. *Escuchar*: una vez seleccionado un determinado fichero, este botón hará que se reproduzca el audio correspondiente al fichero que se esté mostrando por pantalla en ese momento.

6. *Panel de mensajes*: sobre este panel se mostrarán los distintos mensajes informativos o de error, resultado de la ejecución de cualquiera de las órdenes ejecutadas sobre la GUI. Cambiará el color de su fondo en función del mensaje que se muestre. Si se trata de un comando ejecutado correctamente, el fondo será verde. Si el mensaje indica que se está procesando alguna operación, el color mostrado será el amarillo. Si el mensaje corresponde a un error, el fondo será rojo.

7. *Vectores soporte encontrados*: en este panel aparecerá una lista de todos los SVs que hayan sido ubicados en el fichero solicitado. Se mostrará el tipo de fonema y el instante temporal en el que se encuentra. En caso de que en el panel de configuración correspondiente al punto 1 se hubiera seleccionado la opción de mostrar únicamente los SVs de una clase en concreto, en este panel de SVs encontrados sólo aparecerán los elementos de la clase solicitada.

8. *Lista de Fonemas*: se trata de un panel estático, que nos muestra permanentemente la lista de los fonemas que componen el diccionario, emparejados con la clase que los representa.

9. *Panel de representaciones*: sobre esta zona se muestran las distintas opciones de representación gráfica que se ofrecían en el panel de configuración (punto 3).
   Para el caso de que la opción de representación escogida fuera 'Espectrograma' o 'Forma de Onda', la imagen se mostraría sobre la

totalidad de este panel. En el otro caso (opción 'Ambas') el panel se divide en dos representando en la parte superior la forma de onda y en la inferior el espectrograma.

10. *Barra de Herramientas de Matlab*: este elemento es propio de todas las representaciones gráficas de Matlab. Se ha decidido conservarlo para aportar al usuario de la interfaz de las opciones de 'Guardado', 'Impresión', 'Zoom' y 'Movimiento de la imagen', las cuales pueden resultar bastante útiles en determinados análisis.

Selección de Opciones:

Dentro de un mismo panel, las opciones de selección son excluyentes, de forma que sólo podrá estar marcada un única opción en cada momento.

Refresco de Datos:

Si se pulsa el botón *'Dibujar'* (punto 4) y a continuación se modifica alguna de las opciones de representación, los cambios no tendrán efecto hasta que no se pulse de nuevo dicho botón.

Representación de SVs y transiciones:

La herramienta, en caso de que existan, mostrará todos los SVs ubicados en el fichero seleccionado. Estos SVs aparecerán dibujados como una línea vertical, continua y de color negro, que irá desde la parte inferior del *'Panel de Representaciones'*, hasta la parte superior del mismo. Por otro lado, en todos los ficheros que se muestren por pantalla aparecerán las transiciones entre los distintos fonemas. Éstas se representan como líneas verticales, discontinuas y de color rosa, que al igual que los SVs irán desde la parte inferior hasta la superior del *'Panel de Representaciones'*.

C. Código.

Las funciones definidas para el desarrollo de la Herramienta Gráfica se relacionan según el siguiente diagrama de flujo:

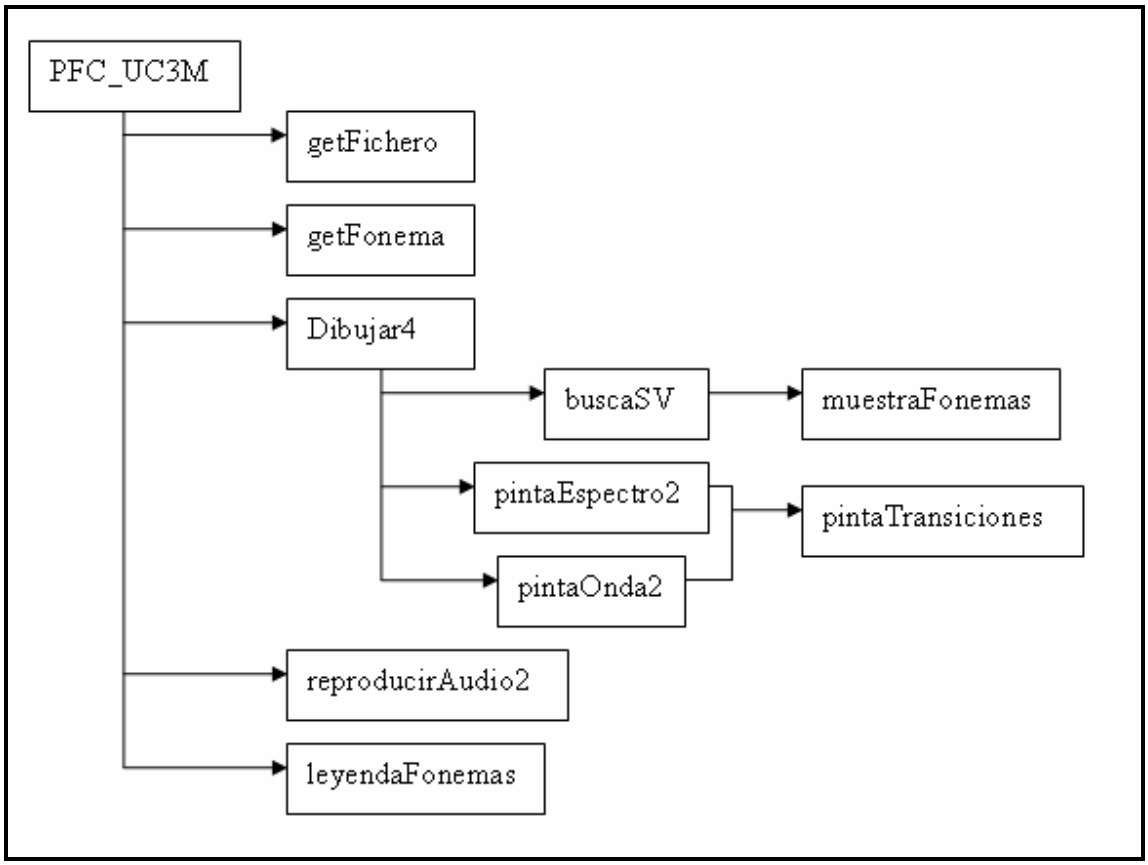


Figura 19. Organización entre funciones del código

A continuación se muestra el código que implementa cada una de las funciones:

- PFC_UC3M.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                             %
%% PFC_UC3M: secuencia de comandos, que crean una interfaz gráfica para       %
%%  representar los SV encontrados.                                            %
%%                                                                             %
%% PARAMETROS: no tiene                                                        %
%%                                                                             %
%% SALIDA: no devuelve ninguna variable, su salida es una interfaz gráfica de %
%%  usuario, que nos permite representar y filtrar los resultados obtenidos.  %
%%                                                                             %
%% FUNCIONES RELACIONADAS:                                                     %
%%  -getFonema                                                                 %
%%  -getFichero                                                                %
%%  -dibujar4                                                                  %
%%  -reproducirAudio2                                                          %
%%  -leyendaFonemas                                                            %
%%                                                                             %
%% AUTOR: Emilio de Oteo (Universidad Carlos III de Madrid)                   %
%%                                                                             %
%% VERSION: 7.0                                                                %
%%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load resul_tot.mat
load lista_nombres.mat
path('C:\Documents and Settings\Emilio\Escritorio\Nueva carpeta\PFC-
EOP\bbdd_audio_mat\', path)
path('C:\Documents and Settings\Emilio\Escritorio\Nueva carpeta\PFC-
EOP\transiciones_mat\', path)

fichero=-1;     %almacena el número del fichero que se quiere mostrar
fonema=0;       %almacena el número correspondiente al fonema que se quiere mostrar
graficas=1;     %almacena el número de gráficas que se quieren mostrar

pantalla=get(0,'ScreenSize');
xw=pantalla(3);
yw=pantalla(4);
fig=figure('Position',[xw*(0.5/10) yw/10 xw*(9/10) yw*(8/10)],...
'Numbertitle','off',...
'Name','PFC - Emilio de Oteo - Universidad Carlos III de Madrid',...
'MenuBar','none',...
'ToolBar','figure');

ft_1=uicontrol(gcf,...
'Style','frame',...
'Position',[17 523 176 99]);

txt_tdir=uicontrol(gcf,...
'Style','text', 'String','Seleccionar SV a representar:',...
'FontWeight', 'bold',...
'Position',[19 600 172 20]);

td_sv1=uicontrol(gcf,...
'Style','radio', 'String','Todos los SV',...
'Position',[30 575 150 25],...
'Value',1,...
'CallBack',[...
'set(td_sv1,''Value'',1),'...
'set(td_sv2,''Value'',0),'...
'fonema=0;,'...
'set(edmulti2,''Enable'',''off''),']);

td_sv2=uicontrol(gcf,...
'Style','radio', 'String','Solo SV del Fonema:',...
'Position',[30 550 150 25],...
'CallBack',[...
'set(td_sv2,''Value'',1),'...
'set(td_sv1,''Value'',0),'...
'fonema=-1;,'...
'set(edmulti2,''Enable'',''on''),']);
```

```
edmulti2=uicontrol(gcf,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',10,'FontName','Arial',...
'String','<=33',...
'Enable','off',...
'Position',[92 525 88 25],...
'Max',1,...
'CallBack',['get(edmulti,''String'');',...
'fonema=getFonema(get(edmulti2,''String''));']);

ft_2=uicontrol(gcf,...
'Style','frame',...
'Position',[17 489 176 29]);

txt_2=uicontrol(gcf,...
'Style','text', 'String','Fichero nº:',...
'FontWeight', 'bold',...
'Position',[30 491 60 20]);

edmulti=uicontrol(gcf,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',10,'FontName','Arial',...
'String','<=16000',...
'Position',[92 491 88 25],...
'Max',1,...
'CallBack',['get(edmulti,''String'');',...
'fichero=getFichero(get(edmulti,''String''));']);

ft_3=uicontrol(gcf,...
'Style','frame',...
'Position',[17 365 176 119]);

txt_tdir=uicontrol(gcf,...
'Style','text', 'String','Seleccionar tipo de representación:',...
'FontWeight', 'bold',...
'Position',[30 442 150 40]);

td_sv3=uicontrol(gcf,...
'Style','radio', 'String','Espectrograma',...
'Position',[30 417 150 25],...
'Value',1,...
'CallBack',[...
'set(td_sv3,''Value'',1),'...
'set(td_sv4,''Value'',0),'...
'set(td_sv5,''Value'',0),'...
'graficas=1;']);

td_sv4=uicontrol(gcf,...
'Style','radio', 'String','Forma de Onda',...
'Position',[30 392 150 25],...
'CallBack',[...
'set(td_sv4,''Value'',1),'...
'set(td_sv5,''Value'',0),'...
'set(td_sv3,''Value'',0),'...
'graficas=2;']);

td_sv5=uicontrol(gcf,...
'Style','radio', 'String','Ambos',...
'Position',[30 367 150 25],...
'CallBack',[...
'set(td_sv5,''Value'',1),'...
'set(td_sv3,''Value'',0),'...
'set(td_sv4,''Value'',0),'...
'graficas=3;']);

ft_4=uicontrol(gcf,...
'Style','frame',...
'Position',[17 271 176 31]);

fig2=axes('Position',[0.29 0.06 0.69 0.9],...
'Visible','on',...
'HandleVisibility','on',...
'XGrid','on',...
'YGrid','on');
```

```
fig3=axes('Position',[0.29 0.76 0.69 0.2],...
'Visible','off',...
'HandleVisibility','off',...
'XGrid','on',...
'YGrid','on');
fig4=axes('Position',[0.29 0.06 0.69 0.64],...
'Visible','off',...
'HandleVisibility','off',...
'XGrid','on',...
'YGrid','on');

pbstart=uicontrol(gcf,...
'Style','push',...
'Position',[17 308 86 50],...
'String','Dibujar',...
'FontWeight', 'bold',...
'CallBack', 'dibujar4(graficas, fichero, fonema, resul_tot, lista_nombres, fig2, fig3,
fig4);');

pbplay=uicontrol(gcf,...
'Style','push',...
'Position',[107 308 86 50],...
'String','Escuchar',...
'FontWeight', 'bold',...
'CallBack', 'reproducirAudio2(lista_nombres, fichero);');

ft_5=uicontrol(gcf,...
'Style','frame',...
'Position',[17 20 176 246]);

txt_tdir=uicontrol(gcf,...
'Style','text', 'String','Vectores Soporte encontrados:',...
'FontWeight', 'bold',...
'Position',[19 240 170 24]);

leyendaFonemas();
```

- getFichero.m

```
function [fichero] = getFichero(fich)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                            %
%% FUNCION getFichero: función que transforma los caracteres que se le        %
%%  pasan por parámetro en un valor numérico y evalúa su validez:             %
%%  [0<=valor<=16000].                                                        %
%%                                                                            %
%% PARAMETROS:                                                                %
%%  -fich: cadena de caracteres a convertir                                   %
%%                                                                            %
%% SALIDA:                                                                    %
%%  -fichero: valor numérico correspondiente a los caracteres que se le       %
%%      pasan por parámetro, en caso de no ser un número válido, devolverá    %
%%      un -1.                                                                %
%%                                                                            %
%% AUTOR: Emilio de Oteo (Universidad Carlos III de Madrid)                   %
%%                                                                            %
%% VERSION: 1.0                                                               %
%%                                                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fichero=-1;
if isequal(fich,'<=16000')==0
    fichero=str2num(fich);
    if fichero < 1
        fichero = -1;
    elseif fichero > 16000
        fichero = -1;
    end
end
if isempty(fichero)==1
    fichero=-1;
end
```

- getFonema.m

```
function dibujar4(num_graf, fichero, fonema, resultados, lista_nombres, fig2, fig3,
fig4)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                               %
%% FUNCION dibujar2: comprueba que el fichero que se quiere representar         %
%%  tiene SV asociados correspondientes al fonema seleccionado.                  %
%%                                                                               %
%% PARAMETROS:                                                                  %
%%  -num_graf: indica si lo que se quiere representar es el espectrograma       %
%%      (1) o la forma de onda de la señal de voz (2).                           %
%%  -fichero: número que representa el fichero que se quiere representar.        %
%%  -fonema: indica el fonema(s) al que pertenecen los SV que se quieren         %
%%      representar.                                                             %
%%  -resultados: matriz con los datos de todos los SV del modelo.                %
%%  -lista_nombres: matriz con los nombres de todos los ficheros ordenados       %
%%      alfabéticamente.                                                         %
%%                                                                               %
%% SALIDA: Esta funcion no devuelve ninguna variable como resultado, su          %
%%  salida es la representación gráfica correspondiente a los datos que se       %
%%  pasan por parámetro.                                                         %
%%                                                                               %
%% FUNCIONES RELACIONADAS:                                                      %
%%  -buscaSV                                                                     %
%%  -pintaEspectro2                                                              %
%%  -pintaOnda2                                                                  %
%%                                                                               %
%% AUTOR: Emilio de Oteo (Universidad Carlos III de Madrid)                      %
%%                                                                               %
%% VERSION: 4.0                                                                 %
%%                                                                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tiempo=clock;
diraudio='C:\Documents and Settings\Emilio\Escritorio\Nueva carpeta\PFC-
EOP\bbdd_audio_mat\';

if fichero == -1

    txt_err1=uicontrol(gcf,...
        'Style','text',...
        'String','Introduzca número de fichero entre 1 y 16000',...
        'BackgroundColor', 'r',...
        'ForegroundColor', 'k',...
        'Position',[20 274 170 25]);
elseif fonema == -1

    txt_err2=uicontrol(gcf,...
        'Style','text',...
        'String','Introduzca número de fonema entre 1 y 33',...
        'BackgroundColor', 'r',...
        'ForegroundColor', 'k',...
        'Position',[20 274 170 25]);
else

    txt_err3=uicontrol(gcf,...
        'Style','text',...
        'String','Espere...',...
        'BackgroundColor', 'y',...
        'ForegroundColor', 'k',...
        'Position',[20 274 170 25]);
    pause(1)
    ft_5=uicontrol(gcf,...
        'Style','frame',...
        'Position',[17 20 176 246]);
    txt_tdir=uicontrol(gcf,...
        'Style','text', 'String','Vectores Soporte encontrados:',...
        'FontWeight', 'bold',...
        'Position',[19 240 170 24]);

    [posicion, longitud]=buscaSV(fonema, fichero, resultados, fig2, fig3, fig4);
```

```
    if isempty(posicion)~=1
        n_fich=lista_nombres(fichero, :);
        %ruta=strcat(diraudio,n_fich,'.mat');
        %!cp eval(ruta) fichero.mat
        %load fichero.mat
        %path('C:\Documents and Settings\Emilio\Escritorio\Nueva carpeta\PFC-
EOP\bbdd_audio_mat\', path)
        sentencia1=strcat('load a',n_fich(2:10),'.mat');
        eval(sentencia1);
        sentencia2=strcat('voice = a',n_fich(2:10),';');
        eval(sentencia2);
        sentencia3=strcat('load t_',n_fich(1:10),'.mat');
        eval(sentencia3);
        sentencia4=strcat('transiciones = t_',n_fich(1:10),';');
        eval(sentencia4);

        if num_graf == 1
            set(fig2, 'HandleVisibility', 'off');
            set(fig3, 'HandleVisibility', 'off');
            set(fig4, 'HandleVisibility', 'off');
            set(fig3, 'HandleVisibility', 'on');
            cla
            set(fig3, 'HandleVisibility', 'off');
            set(fig4, 'HandleVisibility', 'on');
            cla
            set(fig4, 'HandleVisibility', 'off');
            set(fig3, 'Visible', 'off');
            set(fig4, 'Visible', 'off');
            set(fig2, 'Visible', 'on');
            set(fig2, 'HandleVisibility', 'on');
            cla
            pintaEspectro2(posicion, longitud, voice, transiciones);
            %plot(sin(1:1:50));
        end

        if num_graf == 2
            set(fig2, 'HandleVisibility', 'off');
            set(fig3, 'HandleVisibility', 'off');
            set(fig4, 'HandleVisibility', 'off');
            set(fig3, 'HandleVisibility', 'on');
            cla
            set(fig3, 'HandleVisibility', 'off');
            set(fig4, 'HandleVisibility', 'on');
            cla
            set(fig4, 'HandleVisibility', 'off');
            set(fig3, 'Visible', 'off');
            set(fig4, 'Visible', 'off');
            set(fig2, 'Visible', 'on');
            set(fig2, 'HandleVisibility', 'on');
            cla
            pintaOnda2(posicion, longitud, voice, transiciones);
            set(fig2, 'XLim', [0, longitud/100]);
            %pintaSV3(posicion, longitud, fichero, lista_nombres);
            %pintaSV2(resultados, 12, lista_nombres);
        end

        if num_graf == 3
            set(fig2, 'HandleVisibility', 'off');
            set(fig3, 'HandleVisibility', 'off');
            set(fig4, 'HandleVisibility', 'off');
            set(fig2, 'HandleVisibility', 'on');
            cla
            set(fig2, 'HandleVisibility', 'off');
            set(fig2, 'Visible', 'off');
            set(fig3, 'Visible', 'on');
            set(fig4, 'Visible', 'on');
            set(fig3, 'HandleVisibility', 'on');
            cla
            pintaOnda2(posicion, longitud, voice, transiciones);
            set(fig3, 'XLim', [0, longitud/100]);
            set(fig3, 'HandleVisibility', 'off');
            set(fig4, 'HandleVisibility', 'on');
            cla
            pintaEspectro2(posicion, longitud, voice, transiciones);
            set(fig3, 'HandleVisibility', 'on');
        end
```

```
        tiempo=etime(clock, tiempo);
        cadena=strcat('Listo en: ',num2str(tiempo),' seg.');
        txt_err1=uicontrol(gcf,...
            'Style','text',...
            'String',cadena,...
            'BackgroundColor', 'g',...
            'ForegroundColor', 'k',...
            'Position',[20 274 170 25]);

    sentencia3=strcat('clear a',n_fich(2:10));
    eval(sentencia3);

    end
end
```

- reproducirAudio2.m

```
function reproducirAudio2(lista_nombres, fichero)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                            %
%% FUNCION reproducirAudio: reproduce el audio del fichero que se le pasa     %
%%  por parámetro.                                                            %
%%                                                                            %
%% PARAMETROS:                                                                %
%%  -lista_nombres: vector con los nombres de los ficheros de audio           %
%%      ordenados alfabéticamente.                                            %
%%  -fichero: número que representa el fichero que se quiere representar.    %
%%                                                                            %
%% SALIDA: su salida es el audio del fichero.                                 %
%%                                                                            %
%% AUTOR: Emilio de Oteo (Universidad Carlos III de Madrid)                   %
%%                                                                            %
%% VERSION: 2.0                                                               %
%%                                                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tiempo=clock;
diraudio='C:\Documents and Settings\Emilio\Escritorio\Nueva carpeta\PFC-
EOP\bbdd_audio_mat\';
if fichero == -1

    txt_err1=uicontrol(gcf,...
        'Style','text',...
        'String','Introduzca número de fichero entre 1 y 16000',...
        'BackgroundColor', 'r',...
        'ForegroundColor', 'k',...
        'Position',[20 274 170 25]);
else

    txt_err3=uicontrol(gcf,...
        'Style','text',...
        'String','Espere...',...
        'BackgroundColor', 'y',...
        'ForegroundColor', 'k',...
        'Position',[20 274 170 25]);
    pause(1)

    n_fich=lista_nombres(fichero, :);
    sentencia1=strcat('load a',n_fich(2:10),'.mat');
    eval(sentencia1);
    sentencia2=strcat('audio = a',n_fich(2:10),';');
    eval(sentencia2);
    tiempo=etime(clock, tiempo);
    soundsc(audio);

    cadena=strcat('Listo en: ',num2str(tiempo),' seg.');
    txt_err1=uicontrol(gcf,...
            'Style','text',...
            'String',cadena,...
            'BackgroundColor', 'g',...
            'ForegroundColor', 'k',...
            'Position',[20 274 170 25]);

    sentencia3=strcat('clear a',n_fich(2:10));
```

```
    eval(sentencia3);
end
```

- leyendaFonemas.m

```
function leyendaFonemas()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                            %
%% FUNCION leyendaFonemas: representa la lista de clases y su correspondencia %
%%  con los fonemas                                                           %
%%                                                                            %
%% SALIDA: printado en pantalla de la lista de fonemasv                       %
%%                                                                            %
%% AUTOR: Emilio de Oteo (Universidad Carlos III de Madrid)                   %
%%                                                                            %
%% VERSION: 1.0                                                               %
%%                                                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load lista_fonemas.mat

ft_6=uicontrol(gcf,...
    'Style','frame',...
    'Position',[196 20 75 602]);

txt_tfon1=uicontrol(gcf,...
    'Style','text', 'String','Lista de Fonemas:',...
    'FontWeight', 'bold',...
    'Position',[198 590 69 30]);

for i=1:length(lista_fonemas(:,1))
    if i<10
        cadena=strcat(num2str(i),'   : ',lista_fonemas(i,:));
    else
        cadena=strcat(num2str(i),' : ',lista_fonemas(i,:));
    end
    txt_tfon2=uicontrol(gcf,...
        'Style','text', 'String',cadena,...
        'HorizontalAlignment', 'left',...
        'Position',[215 575-(15*i) 55 15]);
end
```

- buscaSV.m

```
function [pos_sv, lon_fich] = buscaSV(fonema, fichero, resultados, fig2, fig3, fig4)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                              %
%% FUNCION buscaSV: comprueba que para el fichero seleccionado existan SV       %
%%  del tipo seleccionado.                                                      %
%%                                                                              %
%% PARAMETROS:                                                                 %
%%  -fonema: indica el fonema(s) al que pertenecen los SV que se quieren        %
%%      representar.                                                            %
%%  -fichero: número que representa el fichero que se quiere representar.       %
%%  -resultados: matriz con los datos de todos los SV del modelo.               %
%%                                                                              %
%% SALIDA:                                                                     %
%%  -pos_sv: indica la linea, dentro de su fichero, en la que se encuentra      %
%%      cada SV que cumpla las condiciones requeridas.                          %
%%  -lon_fich: almacena el número de líneas que tiene el fichero                %
%%      seleccionado.                                                           %
%%                                                                              %
%% FUNCIONES RELACIONADAS:                                                     %
%%  -muestraFonemas                                                             %
%%                                                                              %
%% AUTOR: Emilio de Oteo (Universidad Carlos III de Madrid)                     %
%%                                                                              %
%% VERSION: 2.0                                                                %
%%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

lon_fich=0;
pos_sv=[];
sv_fon=[];
[fil1, col1]=find(resultados(:,2)==fichero);
fil2=fil1;
col2=col1;
if isempty(fil1) == 1
    txt_err4=uicontrol(gcf,...
        'Style','text',...
        'String','Este fichero no tiene SV asociados',...
        'BackgroundColor', 'r',...
        'ForegroundColor', 'k',...
        'Position',[20 274 170 25]);
    set(fig2, 'HandleVisibility', 'off');
    set(fig3, 'HandleVisibility', 'off');
    set(fig4, 'HandleVisibility', 'off');
    set(fig2, 'HandleVisibility', 'on');
    cla
    set(fig2, 'HandleVisibility', 'off');
    set(fig3, 'HandleVisibility', 'on');
    cla
    set(fig3, 'HandleVisibility', 'off');
    set(fig4, 'HandleVisibility', 'on');
    cla
    set(fig4, 'HandleVisibility', 'off');
    return
elseif fonema ~= 0
    [fil2, col2]=find(resultados(fil1,1)==fonema);
    fil1=fil1(fil2);
end

if isempty(fil2) == 1
    txt_err5=uicontrol(gcf,...
        'Style','text',...
        'String','No existen SV de este fonema en este fichero',...
        'BackgroundColor', 'r',...
        'ForegroundColor', 'k',...
        'Position',[20 274 170 25]);
    set(fig2, 'HandleVisibility', 'off');
    set(fig3, 'HandleVisibility', 'off');
    set(fig4, 'HandleVisibility', 'off');
    set(fig2, 'HandleVisibility', 'on');
    cla
    set(fig2, 'HandleVisibility', 'off');
    set(fig3, 'HandleVisibility', 'on');
```

```
    cla
    set(fig3, 'HandleVisibility', 'off');
    set(fig4, 'HandleVisibility', 'on');
    cla
    set(fig4, 'HandleVisibility', 'off');
else
    pos_sv=resultados(fil1, 3);
    lon_fich=resultados(fil1(1), 4);
    sv_fon=resultados(fil1, 1);
end

if isempty(fil1)==0
    muestraFonemas(sv_fon, pos_sv);
end
```

- pintaEspectro2.m

```
function pintaEspectro2(posicion, longitud, voice, transiciones)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                            %
%% FUNCION pintaEspectro2: pinta el espectrograma del vector que se le pasa   %
%%  por parámetro y sus SV asociados.                                         %
%%                                                                            %
%% PARAMETROS:                                                                %
%%  -posicion: indica la posición de los SV que se quieren representar.       %
%%  -longitud: almacena el número de líneas de que consta el fichero .prm     %
%%      del que se obtuvo el SV.                                              %
%%  -voice: vector con los valores de la señal de voz que se quiere           %
%%      representar.                                                          %
%%                                                                            %
%% SALIDA:                                                                    %
%%  su salida es el espectrograma de la señal de voz y sus SV.                %
%%                                                                            %
%% FUNCIONES RELACIONADAS:                                                    %
%%  -pintaTransiciones
%%                                                                            %
%% AUTOR: Emilio de Oteo (Universidad Carlos III de Madrid)                   %
%%                                                                            %
%% VERSION: 2.0                                                               %
%%                                                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%X: posiciones en el eje x de los puntos que definirán las lineas
%verticales que indentificarán los SV
X1=[(posicion./100) (posicion./100)]';
%Y: posiciones en el eje y de los puntos que definirán las lineas
%verticales que indentificarán los SV
Y1=(ones(length(posicion),1)*[0, 4000])';

specgram(voice,800,8000);
hold
line(X1, Y1, 'Color', 'k');
hold
pintaTransiciones(transiciones, 1);
hold
clc
```

- pintaOnda2.m

```
function pintaOnda2(posicion, longitud, voice, transiciones)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                            %
%% FUNCION pintaOnda2: pinta la forma de onda del vector que se le pasa por   %
%%  parámetro y sus SV asociados.                                             %
%%                                                                            %
%% PARAMETROS:                                                                %
%%  -posicion: indica la posición de los SV que se quieren representar.       %
%%  -longitud: almacena el número de líneas de que consta el fichero .prm     %
%%      del que se obtuvo el SV.                                              %
%%  -voice: vector con los valores de la señal de voz que se quiere           %
%%      representar.                                                          %
%%                                                                            %
%% SALIDA:                                                                    %
%%  su salida es la representación gráfica de la señal de voz y sus SV.       %
%%                                                                            %
%% FUNCIONES RELACIONADAS:                                                    %
%%  -pintaTransiciones                                                        %
%%                                                                            %
%% AUTOR: Emilio de Oteo (Universidad Carlos III de Madrid)                   %
%%                                                                            %
%% VERSION: 2.0                                                               %
%%                                                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%normalizo la señal de voz
vo=voice./max(abs(voice));
%normalizo el eje temporal para que represente la duración en segundos de
%la señal de voz
eje=[0:((longitud/100)/length(vo)):(longitud/100)-((longitud/100)/length(vo))];
%X: posiciones en el eje x de los puntos que definirán las lineas
%verticales que indentificarán los SV
X=[(posicion./100) (posicion./100)]';
%Y: posiciones en el eje y de los puntos que definirán las lineas
%verticales que indentificarán los SV
Y=(ones(length(posicion),1)*[1, -1])';

plot(eje, vo)
hold
line(X, Y, 'Color', 'k')
hold
pintaTransiciones(transiciones, 0);
hold
grid on
xlabel('Tiempo');
ylabel('Amplitud Normalizada');
clc
```

- muestraFonemas.m

```
function muestraFonemas(fonemas, posicion)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                           %
%% FUNCION muestraFonemas: presenta por pantalla los fonemas y los instantes  %
%%  de tiempo en los que aparecen.                                           %
%%                                                                           %
%% PARAMETROS:                                                               %
%%  -filas: indica las filas de la matriz de resultados, que corresponden    %
%%      a los SV que se representan.                                         %
%%  -resultados: matriz con los datos de todos los SV del modelo.            %
%%                                                                           %
%% SALIDA: muestra por pantalla la información de los SV seleccionados.      %
%%                                                                           %
%% AUTOR: Emilio de Oteo (Universidad Carlos III de Madrid)                  %
%%                                                                           %
%% VERSION: 2.0                                                              %
%%                                                                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load lista_fonemas.mat
cadena='';

ft_5=uicontrol(gcf,...
    'Style','frame',...
    'Position',[17 20 176 246]);
txt_tdir=uicontrol(gcf,...
    'Style','text', 'String','Vectores Soporte encontrados:',...
    'FontWeight', 'bold',...
    'Position',[19 240 170 24]);

%tiempos=[resultados(filas,1) resultados(filas,3)];
tiempos=[fonemas posicion];
[y, j]=sort(tiempos(:,2));
tiempos2=tiempos(j,:);

for i=1:length(fonemas)
    cadena=strcat(lista_fonemas(tiempos2(i,1),:),' :',num2str(tiempos2(i,2)/100),'
seg.');
    if i<=10
        txt_tdir=uicontrol(gcf,...
            'Style','text', 'String',cadena,...
            'HorizontalAlignment', 'left',...
            'Position',[23 236-(20*i) 82 20]);
    else
        txt_tdir=uicontrol(gcf,...
            'Style','text', 'String',cadena,...
            'HorizontalAlignment', 'left',...
            'Position',[107 236-(20*(i-10)) 82 20]);
    end
end
```

- pintaTransiciones.m

```
function pintaTransiciones(transiciones, tipo_grafico)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                            %
%% FUNCION pintaTransiciones: pinta las transiciones entre fonemas del        %
%%  fichero seleccionado.                                                     %
%%                                                                            %
%% PARAMETROS:                                                                %
%%  -transiciones: vector que incluye los puntos donde se ubican las          %
%%      transiciones.                                                         %
%%  -tipo_grafico: indica para que tipo de gráfico se pintan las transiciones %
%%      si para Espectrograma, Forma de Onda o Ambos.                         %
%%                                                                            %
%% SALIDA:                                                                    %
%%  su salida es la representación gráfica de las transiciones        .       %
%%                                                                            %
%%                                                                            %
%% AUTOR: Emilio de Oteo (Universidad Carlos III de Madrid)                   %
%%                                                                            %
%% VERSION: 2.0                                                               %
%%                                                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

transiciones=transiciones(2:(length(transiciones(:,1))-1), :);
X2=[transiciones(:,3) transiciones(:,3)]';
if tipo_grafico==1
    Y2=(ones(length(transiciones),1)*[0, 4000])'
else
    Y2=(ones(length(transiciones),1)*[-1, 1])'
end

trans=line(X2, Y2, 'Color', 'm');
set(trans, 'LineStyle', ':');
```

Bibliografía

[1] HTK, Cambridge University Engineering Department, disponible en http://htk.eng.cam.ac.uk/

[2] C. Chang, C. Lin. "LibSVM – A Library for Support Vector Machines", disponible en http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[3] F. Casacuberta Nolla. "La lengua española y las nuevas tecnologías: Análisis y síntesis de la señal acústica", Centro Virtual Cervantes: http://cvc.cervantes.es/obref/congresos/sevilla/tecnologias/mesaredon_casacuberta.htm

[4] H.M. Rulot Segovia. "Un algoritmo de inferencia gramatical mediante corrección de errores", Tesis Doctoral. Universidad de Valencia, 1992.

[5] Philips. SpeechMagic. http://www.speechrecognition.philips.com/

[6] Via Voice IBM. http://www-01.ibm.com/software/pervasive/embedded_viavoice/

[7] Nuance. Dragon NaturallySpeeking 10. http://spain.nuance.com/

[8] Telisma, TeliSpeech, www.telisma.com/teliSpeech_key_benefits.html

[9] PerlBox, http://perlbox.sourceforge.net/pbtk/

[10] Sphinx, http://cmusphinx.sourceforge.net/html/cmusphinx.php

[11] L.R. Rabiner, B.H. Juang. "Fundamentals of speech recognition", Prentice Hall, New Jersey, 1993.

[12] L. R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, 77(2); págs. 157-286. Febrero 1989.

[13] A. Moreno. "La lengua española y las nuevas tecnologías: Inteligencia Artificial y legua española", Centro Virtual Cervantes: http://cvc.cervantes.es/obref/congresos/sevilla/tecnologias/mesaredon_moreno.htm

[14] J.R. Dreller, J.G. Proakis, J.H.L. Hansen. "Discrete-Time Processing of Speech Signals", Ed. Macmillan Publishing Company, New York, 1993.

[15] L.E. Baum, T. Petrie, G. Soules, N.Weiss. "A Maximization Technique Ocurring in the Statistical Analysis of Probabilistic Functions of Markov Chains", Annals of Mathematical Statistics, 41(1); págs. 154-171, 1970.

[16] R. Singh. "Diseño de Sistemas ASR basados en HMM". http://mit.ocw.universia.net/6.345/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-345Automatic-Speech-RecognitionSpring2003/71DB5BA6-5C29-45A7-A8AB-8841373120A9/0/lecture14.pdf

[17] S. J. Young, N. H. Russell, J. H. S. Thornton. "Token Passing: A Conceptual Model for Connected Speech Recognition Systems". Technical Report, Cambridge University, 1989.

[18] E. Trentin, M. Gori. "A survey of hybrid ANN/HMM models form Automatic Speech Recognition", Neurocomputing, 37; págs. 91-126, 2001.

[19] N. Morgan, H. Bourlard. "An Introduction to Hybrid HMM/Conectionist Continuous Speech Recognition", IEEE Signal Processing, págs. 22-45, mayo 1995.

[20] C.J.C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, 2(2); págs.121-167, 1998.

[21] V. N. Vapnik. "The nature of Statistical Learning Theory", Springer-Verlag, New York, 1995.

[22] J. Weston, C. Watkins. "Multi-class Support Vector Machines", Technical Report, Department of Computer Science, Royal Holloway, University of London. 1998.

[23] T.-F. Wu, C.-J. Lin, R. C. Weng. "Probability Estimates for Multi-class Classification by Pairwise Coupling". Journal of Machine Learning Research, 5; págs. 975-1005, 2004

[24] Spanish SpeechDat(II) FDB-4000, http://www.elda.org/catalogue/en/speech/S0102.html